Towards Securing Machine Learning Algorithms through Misclassification Detection and Adversarial Attack Detection

Federica Granese

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAX029

Thèse de doctorat

Vers la Sécurisation des Algorithmes d'Apprentissage Automatique par Misclassification Detection et Adversarial Attack Detection

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à École Polytechnique et Sapienza Università di Roma

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 21 Avril 202, par

FEDERICA GRANESE

Composition du Jury :

Marc Schoenauer Directeur de recherche, Inria (TAU)	Président
Emanuele De Cristofaro Professeur, University College London	Rapporteur
Mario Fritz Professeur, CISPA Helmholtz Center for Information Security	Rapporteur
Ismail Ben Ayed Professeur, École de Technologie Superieure (Lio-Livia)	Examinateur
Elena Marchiori Professeur, Radboud Universiteit (iCIS)	Examinateur
Maks Ovsjanikov Professeur, École Polytechnique (GeoViC)	Examinateur
Catuscia Palamidessi Directeur de recherche, Inria, École Polytechnique & IPP, (COMÈTE)	Directeur de thèse
Daniele Gorla Professeur, Sapienza Università di Roma	Co-directeur de thèse
Pablo Piantanida Professeur, CentraleSupélec, International Laboratory on Learning Systems CNRS	Invité

Institute Polytechnique de Paris
Thése de Doctorat
Spécialité Informatique

Towards Securing Machine Learning Algorithms through Misclassification Detection and Adversarial Attack Detection

Federica Granese

Rapporteurs:	Emanuele De Cristofaro Mario Fritz
Directeur de thése:	Catuscia Palamidessi
Co-Directeur de thése:	Daniele Gorla
Advisor:	Pablo Piantanida
Examinateurs:	Ismail Ben Ayed Elena Marchiori Maks Ovsjanikov Marc Schoenauer (Président du Jury)

To past adventures and those yet to come, to Montella and Italy where my roots reside, and to the world that I have yet to explore. To the people who have crossed my path, to those who have been a source of inspiration, and to those I have yet to meet. May we always strive to take a step forward and never step back.

Abstract

Deep Neural Networks (DNNs) have seen significant advances in recent years and are nowadays widely used in a variety of applications. When it comes to safety-critical systems, developing methods and tools to make these algorithms reliable, particularly for non-specialists who may treat them as "black boxes" with no further checks, constitutes a core challenge. The purpose of this thesis is to investigate various methods that can enable the safe use of these technologies.

In the first part, we tackle the problem of *identifying whether the prediction of a DNN classifier should (or should not) be trusted* so that, consequently, it would be possible to accept or reject it. In this regard, we propose a new detector which approximates the most powerful (Oracle) discriminator based on the probability of classification error with respect to the true class posterior probability. Two scenarios are investigated: Totally Black Box (TBB), where only the soft-predictions are available and Partially Black Box (PBB) where gradient-propagation to perform input pre-processing is allowed. The proposed detector can be applied to any pre-trained model, it does not require prior information about the underlying dataset and is as simple as the simplest available methods in the literature.

We address in the second part the problem of *multi-armed adversarial attacks detection*. The detection methods are generally validated by assuming a single implicitly known attack strategy, which does not necessarily account for real-life threats. Indeed, this can lead to an overoptimistic assessment of the detectors' performance and may induce some bias in comparing competing detection schemes. We propose a novel multi-armed framework for evaluating detectors based on several attack strategies to overcome this limitation. Among them, we make use of three new objectives to generate attacks. The proposed performance metric is based on the worst-case scenario: detection is successful if and only if all different attacks are correctly recognized. Moreover, following this setting, we for-

mally derive a simple yet effective method to aggregate the decisions of multiple trained detectors, possibly provided by a third party. While every single detector tends to underperform or fail at detecting types of attack that it has never seen at training time, our framework successfully aggregates the knowledge of the available detectors to guarantee a robust detection algorithm. The proposed method has many advantages: it is *simple* as it does not require further training of the given detectors; it is *modular*, allowing existing (and future) methods to be merged into a single one; it is *general* since it can simultaneously recognize adversarial examples created according to different algorithms and training (loss) objectives.

Résumé

Les réseaux de neurones profonds ont connu des progressions significatives ces dernières années et sont aujourd'hui largement utilisés dans une variété d'applications. Lorsqu'il s'agit de systèmes critiques pour la sécurité, le développement de méthodes et d'outils pour rendre ces algorithmes fiables constitue un défi central, en particulier pour les non-spécialistes qui peuvent les traiter comme des "boîtes noires" sans autre vérification. L'objectif de cette thèse est d'étudier différentes méthodes qui peuvent permettre l'utilisation sécuritaire de ces technologies.

D'abord, nous devons identifier si la prédiction d'un classificateur devrait (ou ne devrait pas) être fiable afin que il soit possible de l'accepter ou de la rejeter. A cet égard, nous proposons un nouveau détecteur qui approxime le discriminateur le plus puissant (Oracle) basé sur la probabilité d'erreur de classification calculée par rapport à la vraie probabilité postérieure du classificateur. Deux scénarios sont étudiés : Totally Black Box (TBB), où seules les soft-predictions sont disponibles et Partially Black Box (PBB) où la propagation du gradient est autorisée pour effectuer le input pre-processing. Le détecteur proposé peut être appliqué à n'importe quel modèle pre-trained, il ne nécessite pas d'informations préalables sur le dataset et est aussi simple que les méthodes les plus basiques disponibles dans la littérature.

Nous poursuivons en abordant le problème de multi-armed adversarial example detection. Les méthodes de détection sont généralement validées en supposant une seule stratégie d'attaque implicitement connue, ce qui ne réalise pas nécessairement des menaces réelles. En effet, cela peut conduire à une évaluation trop optimiste des performances des détecteurs et peut induire un certain biais dans la comparaison des schémas de détection concurrents. Nous proposons un nouveau framework multi-armed pour évaluer les détecteurs sur la base de plusieurs stratégies d'attaques afin de surmonter cette limitation. Parmi celles-

ci, nous utilisons trois nouvelles fonctions objectifs pour générer des attaques. La mesure de performance proposée est basée sur le scénario du worst case : la détection est réussie si et seulement si toutes les différentes attaques sont correctement reconnues. De plus, en suivant ce framework nous dérivons formellement une méthode simple mais efficace pour agréger les décisions de plusieurs détecteurs entraînés éventuellement fournis par une tierce partie. Alors que chaque détecteur a tendance à sous-performer ou à échouer dans la détection de types d'attaques qu'il n'a jamais vus au moment de l'entraînement, notre framework permet d'agréger avec succès les connaissances des détecteurs disponibles pour garantir un algorithme de détection robuste. La méthode proposée présente de nombreux avantages : elle est simple car elle ne nécessite pas d'entraînement supplémentaire des détecteurs donnés ; elle est modulaire, permettant aux méthodes existantes (et futures) d'être fusionnées en une seule ; elle est générale car elle peut reconnaître simultanément des exemples adverses créés selon différents algorithmes et objectifs d'entraînement.

Acknowledgements

I express my deepest gratitude and appreciation to all those who have supported and guided me throughout my doctoral journey and the completion of this thesis.

First, I would like to thank my advisor, Prof. Pablo Piantanida, for his unwavering guidance, expertise, and continuous support throughout this research endeavor. His valuable insights, constructive feedback, and encouragement have been instrumental in shaping this work.

I am deeply grateful to Prof. Catuscia Palamidessi who financed me through the HYPATIA project of the European Research Council (ERC) under the Horizon 2020 research and innovation program of the European Union (Grant agreement N. 835294). Financial assistance has played a crucial role in the execution of this research work.

I would like to underline my gratitude to Prof. Daniele Gorla, with whom I started this journey in the research world nearly six years ago and who enabled me to embark on my academic path in France.

I acknowledge the reviewers of this thesis, Prof. Emanuele De Cristofaro and Prof. Mario Fritz, for their feedback and evaluation, which have helped improve the quality of this work. I also extend my appreciation to all the members of the jury. The questions posed during the evaluation have sparked my curiosity to extend this work into new areas, and I believe this is the essence of research – to constantly explore and push the boundaries of knowledge.

I would like to express a special thanks to Dr. Marco Romanelli, who has been not only a colleague but also a friend. The works presented in this thesis have emerged from a fruitful collaboration with him. I am grateful for the stimulating discussions and for his encouragement during times when my motivation wavered due to unexpected experimental results. Marco, along with Pablo, has been a true mentor to me.

I extend my sincere appreciation to the entire COMÈTE team at Inria Saclay for fostering a stimulating academic environment. The collaborative opportunities and academic discussions within the team have been invaluable to my research journey. I am grateful for the friendships I have formed within the team. In particular, I would like to express my gratitude to Sayan Biswas, Carlos Pinzon, Santiago Quintero, Sergio Ramirez, Ganesh del Grosso, Renan Spencer Trindade, and Fabricio Morales for their support. Each of them has played a role in my journey. I appreciate the meaningful connections we have formed and the experiences we have shared. Lastly, I would like to thank Ruben Santisteban Salazar, with whom I am rediscovering France and Paris through a different lens and who is making my current days happier.

Finally, I am deeply grateful to my family for their unwavering encouragement, and understanding throughout my academic pursuits. The constant support and belief in my abilities of my parents, Alfonso Granse and Nissia Figliuolo, have been my driving force. I am grateful for the profound reflections and insightful conversations I shared with my sister Libera Granese and Giovanni Recupido. Their guidance and wisdom have not only contributed to my professional growth but also enriched me as an individual. I value their support and the meaningful impact they have had on my personal and professional development.

To everyone who has contributed, directly or indirectly, to this thesis, I express my heartfelt appreciation. Thank you all.

PS: Finally, I would like to thank ChatGPT for helping me write these acknowledgments.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 From the general learning problem to detection 3
1.2.1 The general learning problem 3
1.2.2 Detection as hypothesis testing 4
1.3 Misclassification detection 5
1.4 Multi-armed adversarial attacks detection 7
1.5 Plan of the thesis and contributions 10

2 Preliminaries **13**
2.1 Multiclass classification 13
2.1.1 Basic definitions 13
2.1.2 Error variable 14
2.2 Attacking neural networks 15
2.2.1 Adversarial problem 15
2.2.2 Why do adversarial examples exist? 16
2.2.3 Crafting adversarial attacks 18
2.2.4 Protecting from adversarial attacks 19

I Misclassification Detection **21**

3 DOCTOR: A Simple Method for Detecting Misclassification Errors **23**
3.1 The Optimal Discriminator 24
3.1.1 Statistical model for detection 24
3.1.2 Performance metrics and optimal discriminator 24

3.2 The Proposed Discriminator: DOCTOR 26
3.2.1 DOCTOR discriminator 26
3.2.2 Discussion 27
3.2.3 From the theory to a practical discriminator 27
3.3 Evaluation 28
3.4 Experimental Results 28
3.4.1 Review of related methods 29
3.4.2 Detection of misclassification errors, experimental setup and evaluation metrics 31
3.4.3 Experimental results: comparison between different discriminators 33
3.5 Final remarks 36

II Multi-Armed Adversarial Attack Detection 39

4 MEAD 41
4.1 Generating adversarial examples according to different objectives . 42
4.2 A case study: ACE vs. Gini Impurity 43
4.3 Evaluation with a Multi-Armed Attacker 45
4.4 Experiments 46
4.4.1 Experimental setting 46
4.4.2 Experimental results 49
4.5 Final remarks 52

5 A Minimax Approach Against MEAD 53
5.1 Formalization of the Problem of Detecting multi-armed Adversarial Attacks 54
5.1.1 Statistical model 54
5.1.2 A novel objective for detection under simultaneous attacks 54
5.2 Experimental Results 58
5.2.1 Evaluation framework 58
5.2.2 Discussion 61
5.3 Final remarks 67

6 Conclusion of the Thesis **71**

References

A Appendix to Chapter 3

A.1 Proofs . . . .
A.1.1 Proof of Proposition 1 . . . .
A.1.2 Proof of Proposition 2 . . . .
A.1.3 Proof of Inequalities in (2.3) . . . .
A.2 Logistic Regression and Gaussian Model . . . .
A.2.1 Theoretical analysis . . . .
A.2.2 Experiments . . . .
A.3 Supplementary Results of Section 3.4 . . . .
A.3.1 Experimental environment . . . .
A.3.2 On the input pre-processing in DOCTOR . . . .
A.3.3 On the effect the intervals considered for γ, δ and ζ have on the AUROC computation . . . .
A.3.4 Additional plots and results . . . .
A.3.5 DOCTOR for pure OOD detection . . . .
A.3.6 Some observations on the white-box scenario (WB) . . . .

B Appendix to Chapter 4

B.1 Additional results . . . .
B.1.1 Additional Results on CIFAR10 . . . .
B.1.2 Additional Results on MNIST . . . .

C Appendix to Chapter 5

C.1 On the optimization of Eq. (5.5) . . . .
C.2 Supplementary Results of Section 5.2 . . . .
C.2.1 On the MEAD framework . . . .
C.2.2 The proposed aggregator against the adaptive-attacks in the MEAD scenario . . . .
C.2.3 AutoAttack . . . .
C.2.4 Additional plots . . . .

List of Figures

1.1 General Learning Problem [Vap95] . . . 3
1.2 Misclassification detection . . . 6
1.3 Multi-armed adversarial attacks detection . . . 8

2.1 The effects of FGSM [GSS15] . . . 15

3.1 Detectors' distribution: comparison between DOCTOR and the competitors . . . 32
3.2 ROC curves: comparison between DOCTOR and the competitors . 34

4.1 A case study: ACE vs. Gini Impurity . . . 43
4.2 MEAD workflow . . . 45

5.3 Box-plot: the *shallow* detectors, the aggregator and NSS . . . 61
5.1 Performance of the *shallow* detectors grouped by L_p-norm and perturbation magnitude ε on CIFAR10 . . . 62
5.2 Discrimination performancesx . . . 63

A.1 Toy example: ROCs . . .
A.2 The effect of varying T and ϵ . . .
A.3 Overall results when varying T and ϵ . . .

C.1 Box-plot: the *shallow* detectors, NSS and FS . . .
C.2 The aggregator against the adaptive-attacks under MEAD . . .
C.3 Additional plots . . .

List of Tables

3.1 Misclassification detection: overall results 36
3.2 Misclassification detection in presence of OOD samples 36

4.1 MEAD: summary of the results on CIFAR10 47
4.2 MEAD: summary of the results on MNIST 47
4.3 MEAD: comparison average performance 48

5.1 Group of simultaneous attacks 59
5.2 Comparison between the aggregator and NSS on CIFAR10 and SVHN 65
5.3 Comparison between Ours (the aggregator) and Ours+NSS on CIFAR10 66
5.4 The aggregator and NSS in the non-simultaneous setting 68

A.1 Toy example: accuracy of f_{θ_i} on the test set. . . . . .
A.2 Toy example: AUROCs . . . . . .
A.3 The effects of varying the number of thresholds . . . . . .
A.4 DOCTOR for pure OOD detection . . . . . .
A.5 DOCTOR for OOD detection of samples similar to the in-distribution ones. . . . . .
A.6 White-box setting . . . . . .
A.7 Misclassification detection in presence of OOD samples: overall results . . . . . .

B.1 Average number of successful attacks per natural sample considered in the single-armed setting and MEAD (CIFAR10) . . . . . .
B.2 Performances on *NSS* per objective and in MEAD on CIFAR10 . .

B.3 Performances on *KD-BU* per objective and in MEAD on CIFAR10
B.4 Performances on *LID* per objective and in MEAD on CIFAR10 . .
B.5 Performances on *FS* per objective and in MEAD on CIFAR10 . . .
B.6 Performances on *MagNet* per objective and in MEAD on CIFAR10
B.7 Average number of successful attacks per natural sample considered in the single-armed setting and MEAD (MNIST) . . . . . .
B.8 Performances on *NSS* per objective and in MEAD on MNIST . . .
B.9 Performances on *KD-BU* per objective and in MEAD on MNIST .
B.10 Performances on *LID* per objective and in MEAD on MNIST . . .
B.11 Performances on *FS* per objective and in MEAD on MNIST . . . .
B.12 Performances on *MagNet* per objective and in MEAD on MNIST .

C.1 Comparison between Ours (the aggregator) and Ours+FS on CIFAR10 . . . . . .
C.2 Simultaneous attacks detection: NSS on CIFAR10 . . . . . .
C.3 Simultaneous attacks detection: the aggregator on CIFAR10 . . .
C.4 Simultaneous attacks detection: NSS on SVHN . . . . . .
C.5 Simultaneous attacks detection: the aggregator on SVHN . . . . .
C.6 The aggregator against the adaptive-attacks under MEAD (each detector together with the classifier attacked once at a time) . . .
C.7 The aggregator against the adaptive-attacks under MEAD (all the detectors and the classifier attacked together at the same time) . .
C.8 Comparison between the aggregator and the single detectors (*stronger* version) against the adaptive-attacks . . . . . .
C.9 Aggregator on AutoAttack (MEAD framework) . . . . . .

CHAPTER 1

Introduction

1.1 Motivation

Machine learning (ML) has rapidly become integral to various industries, including healthcare, finance, transportation, and entertainment. The goal is to enable machines to learn from data and make predictions or decisions without explicit instructions using algorithms and statistical models. Even if ML has revolutionized how we process and analyze data, it poses new risks and challenges, particularly as these technologies are quickly being applied to critical systems, such as autonomous driving vehicles or industrial robots, including–but not limited to–classification and decision-making tasks.

Therefore, a major concern with ML is its safety or the possibility of unintended consequences. Developing methods and tools to make these algorithms reliable, particularly for non-specialists who may treat them as "black boxes" with no further checks, constitutes a core challenge. Some of the risks associated with using ML technologies include bias and discrimination in decision-making, vulnerabilities to cyber-attacks, and unintended outcomes due to poorly designed or trained algorithms: according to the Washington Post, in 2021, Tesla vehicles equipped with autopilot software were involved in 273 reported crashes, some of which were fatal[1]; in 2014 the Amazon AI-based experimental hiring tool was discovered to be biased against women[2]; in 2021 IBM's Watson started provid-

[1] https://www.washingtonpost.com/technology/2022/06/15/tesla-autopilot-crashes/

[2] https://www.reuters.com/article/amazon-com-jobs-automation/insight-amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idINKCN1MK0AH?edition-redirect=in

ing incorrect and several unsafe recommendations for the treatment of cancer patients[3].

Researchers and practitioners have made several efforts to improve the safety of ML systems, including creating transparent and interpretable ML models, devising resilient and secure algorithms, and considering ethical concerns during the development and implementation of ML systems. In particular, Hendrycks et al. [HCSS21] categorize the study of machine learning (ML) safety into four main areas. *Robustness*, involving both long-tail and adversarial robustness. Long-tail robust models are required to be resilient to long-tail events, i.e., events that are harder to predict (e.g., 9/11, the financial crisis of 2008, and COVID-19), and generally result in catastrophic outcomes. These events are characterized by long-tail distributions, i.e., probability distributions whose tails are not exponentially bounded [Asm03]. On the other hand, adversarial robustness requires models to be resilient to carefully crafted and deceptive threats rather than unpredictable ones. *Monitoring* highly correlated with anomaly detection, i.e., detecting data instances that significantly deviate from the majority of data instances [PSCvdH22]. One research topic in which anomaly detection is actively studied is out-of-distribution detection (OOD) [HMD19], where the main goal is to prevent errors by identifying potential drifts of the testing distribution. Monitoring also concerns confidence calibration, i.e., the problem of predicting probability estimates representative of the true correctness likelihood [GPSW17]. *Alignment* which, as the name implies, refers to aligning the objective functions used to drive system behavior with human values. For instance, one definition of fairness in machine learning considers discrimination against a specified sensitive attribute in supervised learning [HPS16]. *Systematic safety*, which includes cybersecurity and tools to help decision-makers handle ML systems in highly uncertain, quickly evolving, turbulent situations [HCSS21].

[3] https://www.nytimes.com/2021/07/16/technology/what-happened-ibm-watson.html

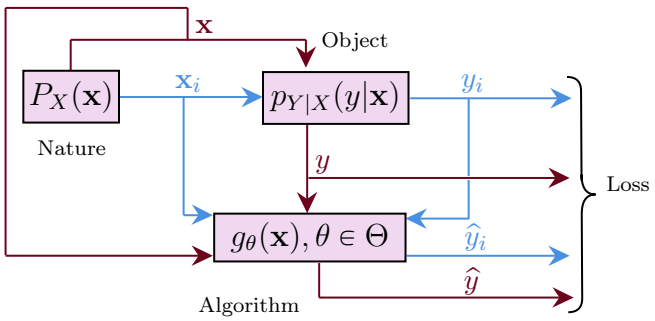


Figure 1.1: General Learning Problem [Vap95]. In light blue is the training phase, and in red is the testing phase.

1.2 From the general learning problem to detection

1.2.1 The general learning problem

To help the comprehension of the next sections, we briefly recall in Fig. 1.1 the *general learning problem* [Vap95]. As shown in Fig. 1.1, the general model consists of three components: (i) the *nature* namely the fixed but unknown distribution p_X over a $\mathcal{X} \subseteq \mathbb{R}^d$, where $\mathcal{X}$ is also known as input space; (ii) the *object* namely the fixed but unknown conditional distribution $p_{Y|X}$ which returns an output value $y \in \mathcal{Y}$ (the concept) to every input vector $\mathbf{x} \in \mathcal{X}$, where $\mathcal{Y}$ is the label space that can be either discrete or continuous; (iii) the *learning machine* capable of implementing a set of functions $g_\theta(\mathbf{x}), \theta \in \Theta$, where Θ is a set of parameters. Classical machine learning aims to select the function that best approximates the object's output: given a training set of n i.i.d. observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ drawn according to $p_{XY} = p_X p_{Y|X}$, we wish to identify a function g_θ, namely the predictor (e.g., neural network), minimizing the empirical risk, i.e., the discrepancy between the object's response y to the input $\mathbf{x}$ and the response provided by the learning machine. Based on this approach, if the object's output and the learning machine algorithm's output are highly likely to be the same, we say the predictor performs well. In general, this is called the *philosophical instrumentalism approach* (imitation of the object): science's role is to predict the concept, regardless of the law of nature. This is slightly different from the *philosophical realism approach* (approximation of the object) where science's role is to try to approximate the object itself.

1.2.2 Detection as hypothesis testing

After addressing the initial learning problem depicted in Fig. 1.1, we now proceed to the next phase by creating a decision rule that employs the acquired distribution $p_{\hat{Y}|X}$ to resolve a specific *detection* task.

Detection problems are usually cast as binary (or M-ary) hypothesis testing. Suppose it is given to us a random variable (r.v.) X following one of the two probability density functions (pdf) p_0 and p_1 defined on a finite set $\mathcal{X}$. The true distribution is unknown to us, but we want to distinguish between the following two hypotheses, i.e., $\mathbb{H}_0$: $X \sim p_0 \equiv p_{X|H}(\cdot|0)$ (also known as *null* hypothesis); $\mathbb{H}_1$: $X \sim p_1 \equiv p_{X|H}(\cdot|1)$. The objective of binary hypothesis testing is then to develop a decision rule $d : \mathcal{X} \to \{0, 1\}$ for making the best guess about which hypothesis is correct. This context can lead to four possible outcomes. Generally, when $\mathbb{H}_1$ is true, and we choose it, we refer to that situation as a *detection*. Likewise when $\mathbb{H}_0$ is true but we select $\mathbb{H}_1$ we call that a *false alarm* (*Type-I* error); when $\mathbb{H}_1$ is true but we choose $\mathbb{H}_0$ we call that a *miss* (*Type-II* error). The decision rule will partition the input space $\mathcal{X}$, into two regions: $\mathcal{X}_0$ where the observations are consistent with $\mathbb{H}_0$; $\mathcal{X}_1$ where the observations are consistent with $\mathbb{H}_1$. In particular, if we characterize the probability of detection of d as

$$P_D(d) \stackrel{\text{def}}{=} \Pr[\mathcal{X}_1|\mathbb{H}_1] = \int_{x\in\mathcal{X}_1} P_1(x)dx$$

and, likewise, the probability of false alarm of d as

$$P_F(d) \stackrel{\text{def}}{=} \Pr[\mathcal{X}_1|\mathbb{H}_0] = \int_{x\in\mathcal{X}_1} P_0(x)dx,$$

we ideally would like to have $P_D(d) \to 1$ and $P_F(d) \to 0$. Clearly, the main aspiration would be to derive a decision rule splitting the input space optimally, e.g., if we suppose $\Pr[\mathbb{H}_0]$ and $\Pr[\mathbb{H}_1]$ to be known (a priori probabilities), the optimal decision rule will be, for every $x \in \mathcal{X}$:

$$\begin{aligned} p_{H|X}(1|x) &\underset{\mathcal{X}_0}{\overset{\mathcal{X}_1}{\gtrless}} p_{H|X}(0|x) \\ \Pr[\mathbb{H}_1]\, p_{X|H}(x|1) &\underset{\mathcal{X}_0}{\overset{\mathcal{X}_1}{\gtrless}} \Pr[\mathbb{H}_0]\, p_{X|H}(x|0). \end{aligned} \tag{1.1}$$

That is, we assign to $\mathcal{X}_1$ all the samples for which the left side of Eq. (1.1) is greater or equal than the left side; viceversa we assign to $\mathcal{X}_0$ all the samples for which the left side of Eq. (1.1) is smaller than the right side. However, this ideal scenario cannot occur when the two underlying distributions overlap. Thus, to increase the detection probability, we must also allow for the probability of false alarm to increase. This represents the fundamental tradeoff in hypothesis testing and detection theory. We can rewrite Eq. (1.1) as follows

$$\Pr[\mathbb{H}_1]\, p_{X|H}(x|1) \underset{\mathcal{X}_0}{\overset{\mathcal{X}_1}{\gtrless}} \gamma \cdot \Pr[\mathbb{H}_0]\, p_{X|H}(x|0), \tag{1.2}$$

where $\gamma \in \mathbb{R}$ is the threshold regulating the tradeoff. We can assess the performance of the proposed decision rule in terms of *Receiver operating characteristic* (ROC) representing the upper boundary between achievable and un-achievable regions in the (P_F, P_D)-square. The detectors' goal will be to partition the input space accordingly.

The main aim of this thesis is to explore the behavior of neural networks when presented with input samples that are either 'clean' (or 'natural') but may be wrongly classified (*misclassification detection*), as well as samples 'adversarial' that have been intentionally manipulated to deceive the model (*adversarial detection*). Our objective is, therefore, to develop a decision rule that can identify, during testing, when an input sample is likely to cause the network to exhibit a specific behavior and make decisions based on this information.

1.3 Misclassification detection

The scheme in Fig. 1.2 refers to the setting of *misclassification detection*. The goal of the detector is to check whether the prediction made by the classifier is correct and accept or reject it accordingly. Let $E(\mathbf{x}) \overset{\text{def}}{=} \mathbb{1}[Y \neq g_\theta(\mathbf{x})]$ denote the *error variable* for a given $\mathbf{x} \in \mathcal{X}$ with respect to (w.r.t.) g_θ, where $\mathbb{1}[\mathcal{E}]$ is the indicator vector which outputs 1 if $\mathcal{E}$ is true and 0 otherwise. The idea is to model the data distribution as a mixture of two pdfs, one representing the distribution of the wrongly classified samples and the other of the correctly classified samples (see upper left box of Fig. 1.2). More formally, $p_{X|E}(\mathbf{x}|1)$ is the pdf truncated

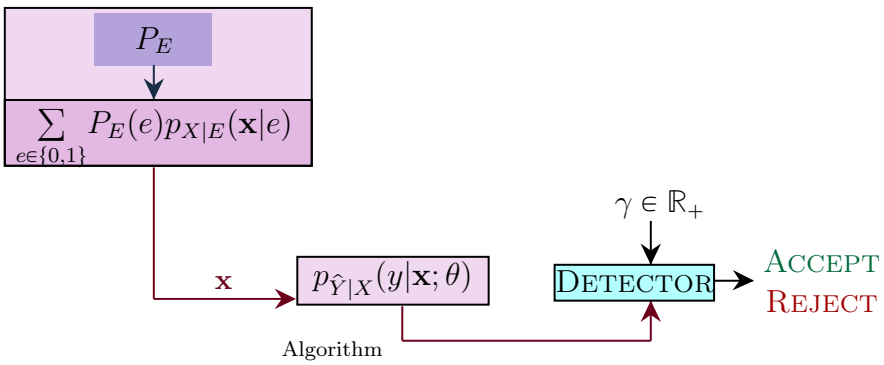


Figure 1.2: Misclassification Detection. DETECTOR in cyan represents either D_α or D_β. The initial learning process is not shown, indicating that it will not be repeated.

to the error event $\{E = 1\}$ (i.e., the hard decision fails) and $p_{X|E}(\mathbf{x}|0)$ is the pdf truncated to the success event $\{E = 0\}$ (i.e., the hard decision succeeds). The problem is therefore cast as in Section 1.2.2 by first identifying our hypothesis

$$\mathbb{H}_0 : X \sim p_{X|E}(\mathbf{x}|0)$$

and

$$\mathbb{H}_1 : X \sim p_{X|E}(\mathbf{x}|1).$$

The most powerful (Oracle) discriminator at threshold $\gamma \in \mathbb{R}$ is defined as

$$D(\mathbf{x}, \gamma) = \begin{cases} 1, & \text{if } p_{X|E}(\mathbf{x}|1) \geqslant \gamma \cdot p_{X|E}(\mathbf{x}|0) \\ 0, & \text{otherwise,} \end{cases} \tag{1.3}$$

where $D(\mathbf{x}, \gamma) = 1$ denotes the sample's prediction is going to be rejected. In Chapter 3, we will show that by applying Bayes theorem Eq. (1.3) can be rewritten in terms of probability of classification error $\mathbf{Pe}(\cdot)$ w.r.t. $p_{Y|X}$. Indeed $p_{E|X}(1|\mathbf{x}) = 1 - p_{Y|X}(g_\theta(\mathbf{x})|\mathbf{x}) = \mathbf{Pe}(\mathbf{x})$ and therefore

$$D(\mathbf{x}, \gamma) = \begin{cases} 1, & \text{if } \mathbf{Pe}(\mathbf{x}) \geqslant \gamma' \cdot (1 - \mathbf{Pe}(\mathbf{x})) \\ 0, & \text{otherwise.} \end{cases} \tag{1.4}$$

To conclude, in Chapter 3 we will present two practical detectors based on the approximation of $\mathbf{Pe}(\mathbf{x})$. We recall the detector in Eq. (1.4) supposes to have

access to all the involved distributions that are typically unknown. Consequently, the practical detectors can rely only on the model posterior distribution $p_{\widehat{Y}|X}$

$$D_\alpha(\mathbf{x}, \gamma) = \begin{cases} 1, & \text{if } \mathrm{Gini}(\mathbf{x}) \geqslant \gamma' \cdot (1 - \mathrm{Gini}(\mathbf{x})) \\ 0, & \text{otherwise.} \end{cases} \tag{1.5}$$

and

$$D_\beta(\mathbf{x}, \gamma) = \begin{cases} 1, & \text{if } \widehat{\mathrm{Pe}}(\mathbf{x}) \geqslant \gamma' \cdot (1 - \widehat{\mathrm{Pe}}(\mathbf{x})) \\ 0, & \text{otherwise.} \end{cases} \tag{1.6}$$

where $\mathrm{Gini}(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{y \in \mathcal{Y}} p_{\widehat{Y}|X}(y|\mathbf{x};\theta)(1 - p_{\widehat{Y}|X}(y|\mathbf{x};\theta))$ is the probability of incorrectly classifying the feature $\mathbf{x}$ if it was randomly labeled according to the model distribution and $\widehat{\mathrm{Pe}}(\mathbf{x}) \stackrel{\text{def}}{=} 1 - p_{\widehat{Y}|X}(g_\theta(\mathbf{x})|\mathbf{x};\theta)$ is the probability of classification error w.r.t. the predicted distribution.

The aforementioned detectors will be evaluated on either image or textual datasets in two alternatives scenario depending on the amount of information about the underlying classifier available. In the *totally black box* (TBB) we suppose only the soft-predictions are available in the *partially black box* (PBB) the gradient-propagation to perform input pre-processing is allowed. In particular, input pre-processing is a common technique used to slightly modify some patterns of an input sample in the direction of an objective loss to maximize/minimize. The concept is the same as when training a neural network where the weights are modified step-by-step to minimize the training loss. In our context, the modification will affect the pixel of the images in the direction where the loss in Eqs. (1.5) and (1.6) maximizes. This technique will show up to be particularly effective for our goal.

1.4 Multi-armed adversarial attacks detection

We recall that adversarial examples are carefully crafted input patterns designed to deceive a target classifier into making an incorrect decision while remaining as similar as possible to the original sample. More formally, let us consider a natural sample, denoted by $\mathbf{x} \in \mathcal{X}$, along with its true label, $y \in \mathcal{Y}$. An attacker aims to

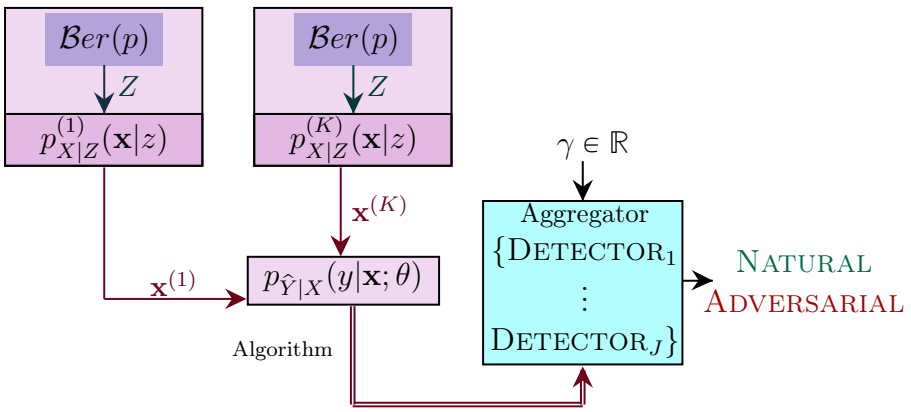


Figure 1.3: Multi-armed adversarial attacks detection. The double arrows mean that the outputs are for each $\mathbf{x}^{(k)}$. Note that $K, J \in \mathbb{N}$ but J can be different from K. $\mathcal{B}er(p)$ denotes the Bernoulli distribution of parameter $p \in [0, 1]$. The initial learning process is not shown, indicating that it will not be repeated.

deceive the model g_θ by crafting an adversarial example, $\mathbf{x}'_\ell \in \mathcal{I} \subseteq \mathbb{R}^d$, where $\mathcal{I}$ is a held-out set of images that is distributed according to p_{XY} but that was not used during training. The symbol ℓ denotes the objective loss function $\ell(\mathbf{x}, \mathbf{x}'_\ell; \theta)$ optimized by the attacker; ε is perturbation magnitude, and L_p, $p \in \{1, 2, \infty\}$ is the norm constraint. The goal of the attack is to obtain an $\mathbf{x}'_\ell$ such that $g_\theta(\mathbf{x}'_\ell) \neq g_\theta(\mathbf{x})$, in order to force the target model to make a prediction error. As thoroughly investigated in [SZS+14], the adversarial generation problem is difficult to tackle and it is commonly relaxed as follows

$$\mathbf{x}_\ell{}' \equiv \mathbf{x}_\ell{}'(\mathbf{x}) = \underset{\mathbf{x}_\ell{}' \in \mathbb{R}^d \,:\, \|\mathbf{x}_\ell{}' - \mathbf{x}\|_p < \varepsilon}{\operatorname{argmax}} \ell(\mathbf{x}, \mathbf{x}_\ell{}'; \theta), \tag{1.7}$$

where $\mathbf{x}'_\ell$ is updated iteration by iteration starting from an initial given value.

The scheme in Fig. 1.3 refers to the setting of *multi-armed adversarial attacks detection*. In this case, the goal of the detector is to check whether the input sample is natural or has been adversarially perturbed according to *some* strategy. We refer to this setting as 'multi-armed' as in the classical detection setting (i.e., 'single-armed') the methods are generally validated by assuming a *single* attack strategy at a time. The proposed setting, as reported in Fig. 1.3, considers the evaluation of multiple instances at the same time. Consequently, in Chapter 4 we suggest an alternative framework for evaluating the performance

of the existing state-of-the-art detectors when the attacks at the evaluation time can be simultaneously crafted according to various algorithms and objective loss functions. The detection will be successful if and only if all different attacks are correctly recognized.

We provide a solution to the problem of multi-armed adversarial attack detection in Chapter 5 by dealing with the following worst-case scenario. Assume it is given a distribution for every attack strategy. We could group all such distributions in a set $\mathcal{M} = \left\{p_{XZ}^{(k)} \,:\, k \in \mathcal{K}\right\}$, where $k \in \mathcal{K}$ represents the index and $\mathcal{Z} = \{0, 1\}$ indicates a binary space label for the adversarial example detection task. At the evaluation time, the attackers select an arbitrary strategy i and then sample an input $\mathbf{x}$ according to $p_{X|Z}^{(i)}(\mathbf{x}|z)$, where $p_{X|Z}^{(i)}(\mathbf{x}|1)$ is the probability density function induced by the chosen attack i and $p_{X|Z}^{(i)}(\mathbf{x}|0) = p_X(\mathbf{x})$ *almost surely* is the probability distribution of the natural samples. The defender is asked to choose between the following two hypotheses

$$\mathbb{H}_0 : X \sim p_{X|Z}^{(k)}(\mathbf{x}|0) \text{ for some } k \in \mathcal{K}$$

and

$$\mathbb{H}_1 : X \sim p_{X|Z}^{(k)}(\mathbf{x}|1) \text{ for some } k \in \mathcal{K}.$$

Moreover, suppose the defender has at her/his disposal a set of soft-detectors one for each of the possible distributions in $\mathcal{M}$,

$$\mathcal{Q} = \left\{q_{\hat{Z}|\mathbf{u}}^{(k)} \,:\, \mathcal{U} \mapsto [0,1]^2\right\}_{k \in \mathcal{K}},$$

with $\mathbf{u} \in \mathcal{U} = \{g_\theta^l(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^d\}$ denotes the space of logits. It is also important to keep in mind that the defender does not know what the attacker's strategy will be. The optimal detector will be the one performing simultaneously well over all the possible attacks in $\mathcal{M}$. In Chapter 5 we will show that this can be formalized as the solution to the following minimax problem

$$\mathcal{L}(\mathcal{Q}, \mathbf{x}) = \min_{q_{\hat{Z}|\mathbf{u}}} \max_{k \in \mathcal{K}} \mathbb{E}_{q_{\hat{Z}|\mathbf{u}}^{(k)}} \left[-\log q_{\hat{Z}|\mathbf{u}}\right], \tag{1.8}$$

where the minimization is performed over all (detectors) distributions $q_{\hat{Z}|\mathbf{u}}$, including elements that are not part of the set $\mathcal{Q}$. Thus, the optimal detector will

be as follows

$$D(\mathbf{x}, \gamma) = \begin{cases} 1, & \text{if } \mathcal{L}(\mathcal{Q}, \mathbf{x}) \geqslant \gamma \\ 0, & \text{otherwise.} \end{cases} \tag{1.9}$$

Notice that, Eq. (1.8) is not tractable computationally. Therefore, in Chapter 5, we show how to derive a surrogate function that can be computationally optimized.

The aforementioned detector will be evaluated in the context of multi-armed adversarial attack detection (as well as in the 'single-armed' setting) by assuming that a third party provides us with four simple supervised detectors (i.e., the detectors in $\mathcal{Q}$) each of them trained to detect a single specific kind of attack. Indeed, in practical setting, besides not knowing the attack strategy, we also do not have access to all the possible detectors which adds to the difficulty.

1.5 Plan of the thesis and contributions

Publications from this dissertation

The content of this dissertation is based on the following publications:

Part I is based on the results presented in **DOCTOR: A Simple Method for Detecting Misclassification Errors** [GRG+21] (Chapter 3), that appeared as Spotlight in the proceedings of the 35th *Conference on Neural Information Processing Systems (NeurIPS2021)*. Federica Granese, Marco Romanelli, Daniele Gorla, Catuscia Palamidessi, Pablo Piantanida.

Part II is based on the results presented in: i) **MEAD: A Multi-Armed Approach for Evaluation of Adversarial Examples Detectors** [GPR+22] (Chapter 4), that appeared in the proceedings of the 33rd *European Conference on Machine Learning and Data Mining (ECMLPKDD2022)*. Federica Granese, Marine Picot, Marco Romanelli, Francisco Messina, Pablo Piantanida. ii) **A Minimax Approach Against Multi-Armed Adversarial Attacks Detection** [GRGP23] (Chapter 5), that has been submitted to the 36th *IEEE Computer Security Foundations Symposium*

(CSF). Federica Granese, Marco Romanelli, Siddharth Garg, Pablo Piantanida.

Other publications

Other works I have contributed to during my PhD (at the time I am writing the thesis):

A) Works in Social Network Privacy/Utility and Differential-Privacy. On the one hand, we studied how to control information propagation in social networks. Users want to communicate and interact freely with their peers. However, if misused, the information they spread can have harmful consequences. There is, therefore, a trade-off between utility, i.e., reaching as many intended nodes as possible, and privacy, i.e., avoiding the unintended ones. In [GGP21], we adapt the basic framework of Backes et al. [BGMS17] to include more realistic features, that in practice influence the way in which information is passed around. More specifically, we consider: (a) the topic of the shared information, (b) the time spent by users to forward information among them, and (c) the user's social behavior. Furthermore, we propose an enhanced formulation of the utility/privacy policies, to maximize the expected number of reached users among the intended ones, while minimizing this number among the unintended ones, and we show how to adapt the basic techniques to these enhanced policies. On the other hand, in cite [GJG+22], we look at the problem of *data protection*. Differential privacy is nowadays one of the best established and theoretically solid tools to ensure data protection. Intuitively, given a set of databases, differential privacy requires that databases that only slightly differ one from the other (e.g., in one individual record) are mapped to the obfuscated values with similar probabilities; this provides privacy to the changed record because statistical functions run on the database should not overly depend on the data of any individual. In this work, we analyze to what extent final users can infer information about the level of protection of their data when the data obfuscation mechanism is a priori unknown to them.

Papers:
- Enhanced models for privacy and utility in continuous-time diffusion

networks, that appeared in the proceedings of the *International Journal of Information Security* [GGP21]. Federica Granese, Daniele Gorla, Catuscia Palamidessi.
- On the (Im)Possibility of Estimating Various Notions of Differential Privacy [GJG+22], that has been submitted to 36th *IEEE Computer Security Foundations Symposium*. Daniele Gorla, Louis Jalouzot, Federica Granese, Catuscia Palamidessi, Pablo Piantanida.

B) Additional works in Machine Learning. On the one hand, we propose a new method (HAMPER) to detect adversarial examples by leveraging the concept of data depths, a statistical notion that provides center-outward ordering of points w.r.t. a probability distribution. In particular, the halfspace-mass (HM) depth exhibits attractive properties such as computational efficiency, which makes it a natural candidate for adversarial attack detection in high-dimensional spaces. Additionally, HM is non-differentiable making it harder for attackers to attack HAMPER via gradient based-methods directly. On the other hand, inspired by the work in [GRG+21], we present a simple yet effective hyperparameter free method to implement the rejection option for a pre-trained classifier. The method is is lightweight since it does not require any re-training of the network, and it is flexible since it can be used with any model that outputs soft-probabilities.

Papers:
- A Halfspace-Mass Depth-Based Method for Adversarial Attack Detection, that has been accepted (with minor revision) to *Transactions on Machine Learning Research (TMLR)*. Marine Picot*, Federica Granese* , Guillaume Staerman, Marco Romanelli, Francisco Messina, Pablo Piantanida, Pierre Colombo.
- Trusting the Untrustworthy: A Cautionary Tale on the Pitfalls of Training-based Rejection Option, that has been submitted to 40th *International Conference on Machine Learning (ICML)*. Eduardo Dadalto Câmara Gomes*, Marco Romanelli* , Federica Granese, Siddharth Garg, Pablo Piantanida.

*Equal contribution.

CHAPTER 2

Preliminaries

This chapter aims to recall the fundamental concepts necessary to understand the thesis' content. We indicate for each section which chapter of the thesis it is related to.

2.1 Multiclass classification

In the following section, we begin by recalling some basic notions of machine learning that will be useful in the course of Chapters 3 and 4.

2.1.1 Basic definitions

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the (possibly continuous) *feature space* and let $\mathcal{Y} = \{1, \dots, C\}$ denote the concept of the *label space* related to some task of interest. We denote by p_{XY} the unknown data distribution over $\mathcal{X} \times \mathcal{Y}$. A *predictor* $g_\theta : \mathcal{X} \to \mathcal{Y}$ uses the inferred model $p_{\hat{Y}|X}(y|\mathbf{x};\theta)$ where $y \in \mathcal{Y}$ and $\theta \in \Theta$ are the learnt parameters,

$$g_\theta(\mathbf{x}) \stackrel{\text{def}}{=} \arg\max_{y \in \mathcal{Y}} \; p_{\hat{Y}|X}(y|\mathbf{x};\theta),$$

and tries to approximate the optimal (Bayes) decision rule $g^\star(\mathbf{x}) \stackrel{\text{def}}{=} \arg\max_{y \in \mathcal{Y}} \; p_{Y|X}(y|\mathbf{x})$. Notice that $p_{\hat{Y}|X}$ can be interpreted as the prediction of the class (label) posterior probability given a sample (e.g., $p_{\hat{Y}|X}(y|\mathbf{x};\theta) \equiv \text{softmax}(\mathbf{x})_y$), while $p_{Y|X}$ is the true (unknown) probability. In several practical scenarios $p_{\hat{Y}|X}$ does not perfectly match $p_{Y|X}$ and still $g_\theta \approx g^\star$ (cf. [GPSW17]).

2.1.2 Error variable

Let $E(\mathbf{x}) \stackrel{\text{def}}{=} \mathbb{1}\left[Y \neq g_\theta(\mathbf{x})\right]$ denote the error variable for a given $\mathbf{x} \in \mathcal{X}$ corresponding to g_θ, i.e., where we denote with $\mathbb{1}[\mathcal{E}]$ the indicator vector which outputs 1 if the event $\mathcal{E}$ is true and 0 otherwise. Similarly, we can define the self-error variable $\widehat{E}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbb{1}\left[\widehat{Y} \neq g_\theta(\mathbf{x})\right]$ also corresponding to the inferred predictor g_θ but based on the prediction model $p_{\widehat{Y}|X}$ of the class posterior probability. Notice that $\widehat{E}(\mathbf{x})$ is observable since the underlying distribution is known. However, $E(\mathbf{x})$ cannot be observed and in general these binary variables do not coincide.

At this stage, it is convenient to introduce the notions of *probability of classification error* for a given $\mathbf{x} \in \mathcal{X}$ with respect to (w.r.t.) both the true class posterior and the predicted probabilities:

$$\mathrm{Pe}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbb{E}\left[E(\mathbf{x})|\mathbf{x}\right] = 1 - p_{Y|X}\left(g_\theta(\mathbf{x})|\mathbf{x}\right), \tag{2.1}$$

$$\widehat{\mathrm{Pe}}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbb{E}\left[\widehat{E}(\mathbf{x})|\mathbf{x}\right] = 1 - p_{\widehat{Y}|X}\left(g_\theta(\mathbf{x})|\mathbf{x};\theta\right). \tag{2.2}$$

Notice that $\widehat{\mathrm{Pe}}(\mathbf{x})$ represents the probability of misclassification of the sample $\mathbf{x}$ with respect to the softmax probability $p_{\widehat{Y}|X}$, which can be interpreted as the model's approximation of nature $p_{Y|X}$. Such approximation is close when the model is well-calibrated. Obviously, $\mathrm{Pe}^\star(\mathbf{x}) \leqslant \mathrm{Pe}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$, where $\mathrm{Pe}^\star(\mathbf{x})$ corresponds to the minimum error of the Bayes classifier: $\mathrm{Pe}^\star(\mathbf{x}) = 1 - p_{Y|X}\left(g^\star(\mathbf{x})|\mathbf{x}\right)$. It is worth mentioning that, by averaging (2.1) over the data distribution, we obtain the error rate of the classifier g_θ. Although $\widehat{\mathrm{Pe}}(\mathbf{x})$ provides a valuable candidate to infer the unknown error variable $E(\mathbf{x})$, it is easy to check that

$$\max\left\{\mathrm{Pe}(\mathbf{x}), \widehat{\mathrm{Pe}}(\mathbf{x})\right\} - \Pr\left(\widehat{Y} = Y|\mathbf{x}\right) \leqslant \Pr\left\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\right\} \leqslant \Pr\left(\widehat{Y} \neq Y|\mathbf{x}\right), \tag{2.3}$$

which in particular implies that the error incurred in using $\widehat{E}(\mathbf{x})$ to predict $E(\mathbf{x})$ is lower bounded by the classification error per sample (2.1) [GRG+21].

Note that, $\mathrm{Pe}(\mathbf{x})$ is not available in practical scenarios and the direct estimation (e.g., based on pairs of inputs and labels) of the true class posterior probability $p_{Y|X}$ cannot be performed. Notice that it is not possible to sample the

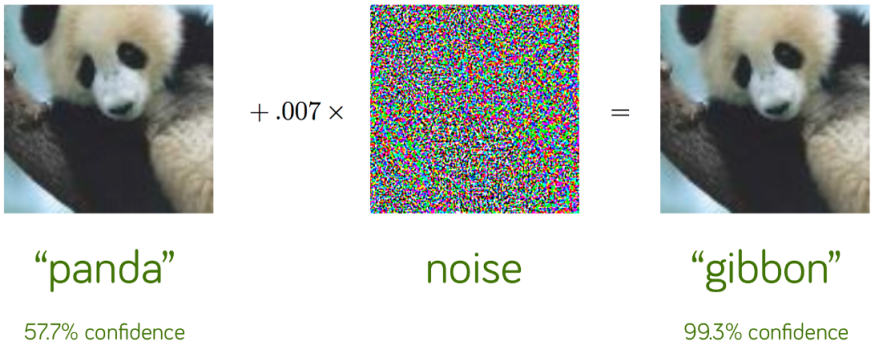


Figure 2.1: The effects of FGSM. The (well known) demonstration of fast adversarial example generation applied to GoogLeNet on ImageNet (from [GSS15]).

conditional pdf $p_{Y|X}$ for each input $\mathbf{x} \in \mathcal{X}$. As a matter of fact, it is well-known that the application of direct methods for this estimation will lead to ill-posed problems, as shown in [VI20].

2.2 Attacking neural networks

In the following section, we provide the background necessary to understand Chapters 4 and 5. Note that this thesis is not fully devoted to studying the adversarial problem. As such, we refer to the survey in [AHFD22] and references therein for a comprehensive discussion of this topic.

2.2.1 Adversarial problem

Adversarial examples were first introduced in 2014 by Szegedy et al. [SZS+14] as a counter-intuitive property of deep neural networks.

Let us consider a natural sample $\mathbf{x} \in \mathcal{X}$ together with its true label $y \in \mathcal{Y}$. An attacker targets the model g_θ by crafting a sample $\mathbf{x}'_\ell \in \mathcal{I} \subseteq \mathbb{R}^d$ according to an objective loss function $\ell(\mathbf{x}, \mathbf{x}'_\ell; \theta)$ which is denoted by ℓ, perturbation magnitude ε, and norm constraint L_p, $p \in \{1, 2, \infty\}$. The goal of the attack is to obtain an $\mathbf{x}'_\ell$ such that $g_\theta(\mathbf{x}'_\ell) \neq g_\theta(\mathbf{x})$, in order to force the target model to make a prediction error. An example is shown in Section 2.2.1).

Formally, Szegedy et al. [SZS+14] define the adversarial generation problem

as:

$$\begin{aligned} &\underset{\mathbf{x}'_\ell \in \mathcal{I}}{\text{argmin}} \; ||\mathbf{x} - \mathbf{x}'_\ell||_p \\ &\text{s.t.} \quad g_\theta(\mathbf{x}'_\ell) \neq y, \end{aligned}$$

where $\mathcal{I}$ is a held-out set of images from the data distribution that the network was not trained. As thoroughly investigated in [MMS+18], the adversarial generation problem as above is difficult to tackle and it is commonly relaxed as follows

$$\mathbf{x}'_\ell \equiv \mathbf{x}'_\ell(\mathbf{x}) = \underset{\mathbf{x}_\ell{}' \in \mathcal{I} \,:\, \|\mathbf{x}'_\ell - \mathbf{x}\|_p < \varepsilon}{\text{argmax}} \ell(\mathbf{x}, \mathbf{x}'_\ell, y; \theta), \tag{2.4}$$

where $\mathbf{x}'_\ell$ is updated iteration by iteration starting from an initial given value and $\ell(\mathbf{x}, \mathbf{x}'_\ell, y; \theta)$ is the objective of the attacker, representing a surrogate of the constraint to fool the target classifier, i.e., $g_\theta(\mathbf{x}'_\ell) \neq y$. The objective function ℓ traditionally used is the Cross-Entropy (CE) [SZS+14, MMS+18]:

$$\ell_{\text{ACE}}(\mathbf{x}, \mathbf{x}'_\ell, y; \theta) = -\log p_{\widehat{Y}|X}(y|\mathbf{x}'_\ell; \theta). \tag{2.5}$$

2.2.2 Why do adversarial examples exist?

Although adversarial examples are easy to grasp, there is much speculation as to why they exist. Below is a brief overview of the most popular hypotheses of the moment. As a starting point, it is worth recalling that natural images are believed to exist in a low-dimensional manifold embedded in a high-dimensional space [CBB19].

Originally, Szegedy et al. [SZS+14] argued that adversarial examples represent densely populated "pockets" in the input space with a low probability of being observed and correctly classified. Later, Goodfellow et al. [GSS15] found the existence of adversarial examples in the linear behavior of classifiers rather than the non-linear behavior of classifiers in high-dimensional spaces.

The *off-manifold assumption* is supported by numerous papers (e.g., [CBB19, FCSG17, MC17] to cite a few), and it is based on the idea that adversarial perturbations push the sample off of the natural data manifold. Interestingly Feinmann et al. [FCSG17] describe three different situations depending on the position of

the adversarial example w.r.t. the decision boundary of the classifier and the submanifold of the adversarial example's class. The adversarial example in the first scenario is close to the decision boundary but far from the submanifolds for the original and new predicted classes; the adversarial example lies in the pocket of the submanifold of the class assigned to the adversarial example in the second scenario; in the third scenario, the adversarial example lies near both the decision boundary and both submanifolds. Note that the off-manifold hypothesis may not be the final explanation on the reason behind the adversarial phenomenon as subsequent papers (e.g. [SHS19, XYF+22]) have shown the existence of on-manifold adversarial examples.

As one of the most widely accepted hypotheses suggests, adversarial examples may be directly related to *non-robust features*, i.e., features that are highly predictive but are brittle, making them incomprehensible to humans. According to this hypothesis, adversarial vulnerabilities are caused by non-robust features, and are not inherently related to standard training frameworks [IST+19]. For example, consider the case of a classifier trained to distinguish *dogs* from *cats*. In addition, suppose the two animals are differentiated based on the direction of the hair rather and the shape itself (features). In this scenario, furry can be considered a non-robust feature, whereas shape can be considered robust. Particularly, the authors demonstrate that training a *non-robust* classifier on the original dataset based on exclusively robust features can achieve both good standard and robust accuracy, which is not the case when training the classifier using non-robust features.

By following a different perspective, the *dimpled manifold model* is a new mental model proposed in 2021 by Shamir et al. [SMB21] in which classifiers place their decision boundaries right next to data manifolds and only slightly curve around them. By going perpendicular to the manifold, adversarial examples can then be found.

Even though there are several possible other explanations for the existence of the adversarial examples (e.g., insufficient data [SST+18] and over-fitting [TG16]), a definitive answer has not yet been found. We would like to highlight that these assumptions are not necessarily mutually exclusive, suggesting (maybe) a unifying theory.

2.2.3 Crafting adversarial attacks

Over the years, a plethora of algorithms to generate adversarial samples has been proposed and, overall, we can group them into two main categories: *white-box* and *black-box* attacks.

White-box attacks

We talk about *white-box* attacks when the adversary knows everything about the target model (its architecture and weights). *Gradient-based* attacks belong to this category. They rely on finding the perturbation direction, i.e., the sign of gradient at each pixel of the input, that maximizes the attacker's objective value. Examples of gradient-based attacks are the *Fast Gradient Sign Method* (FGSM) [GSS15], the *Basic Iterative Method* (BIM) [KGB] and the *Projected Gradient Descent* method (PGD) [MMS+18]. BIM and PGD can be seen as iterative versions of FGSM (one-step perturbation). Unlike BIM, PGD attacks start from a random perturbation in L_p-ball around the input sample. Another powerful attack is the *Carlini-Wagner* attack (CW) [CW17c], which directly minimizes the additive noise constrained by a function which assure the misclassification of the perturbed sample. We conclude the list of white-box attacks by mentioning the *DeepFool attack* (DF) [MFF16], which is an iterative method based on a local linearization of the targeted classifier, and the resolution of the resulting simplified adversarial problem.

Black-box attacks

In the case of *black-box* attacks, the adversary has no access to the internals of the target model, hence it creates attacks by querying the model and monitoring outputs of the model to attack. Examples of black-box attacks are the *Square Attack* (SA) [ACFH20], which iteratively searches for a random perturbation, and checks if it increases the attacker's objective at each step; the *Hop Skip Jump* attack (HOP) [CJW20] which estimates the gradient direction to perturb, and the *Spatial Transformation Attack* (STA) [ETT+19] which transforms the original samples by applying small translations and rotations to them.

It is worth to mention that there also exists *gray-box* attacks, i.e. when the adversary knows the training data but not the internals of the model. These

attacks rely on the transferability property of the adversarial examples: to create attacks these methods build a substitute model that performs the same task as the target model. A special class of attacks are the so-called *adaptive attacks* [ACW18, TCBM20, CW17c, YBTV21] where attacks are specifically designed to target a given defence. In this scenario, the attacker is supposed to have full knowledge of both the targeted classifier and the underlying defence.

2.2.4 Protecting from adversarial attacks

Methods to defend deep models against adversarial attacks can be grouped into two main families: methods that aim to increase the targeted model's robustness by re-training it [GSS15, MMS+18, PMB+22, XTG+20, TKP+18, AF21], and methods engineered to detect adversarial examples at evaluation time [KFHD20, MLW+18, FCSG17, XEQ18, MC17, LLLS18a]. The work in [AHFD22] provides a recent and thorough survey about the state-of-the-art detection methods, which fall under two main categories: *supervised* and *unsupervised*. Below is a brief description of the methods analysed in the thesis.

Supervised methods

Detectors within this category extract features either directly from the targeted network's layer [KFHD20, FCSG17] or by using statistical tools [MLW+18, LLLS18a]. To do so, both natural and adversarial examples are necessary. Generally, the adversarial samples are created according to a single fixed algorithm and a given loss function, which are then also used to create the examples at evaluation time.

Natural Scene Statistics (NSS) [KFHD20]. The approach is based on the hypothesis that natural images possess certain regular statistical properties (i.e., natural scene statistics [SLSZ03, Rud94]) that are altered by adversarial perturbations. Thus, by characterizing these deviations from the regularity of natural statistics using NSS, it possible to determine whether the input is benign or malicious.

Kernel Density and Bayesian Uncertainty (KD-BU) [FCSG17]. Using the intuition that adversarial samples lie off the true data manifold, the method consists in extracting two types of features: *density estimates* that are computed starting from the logits of the network, and *bayesian uncertainty estimates* avail-

able in dropout neural networks. The first metric is meant to check how far an input sample is from the original manifold; the second one to identify if the input sample lies in low-confidence regions of the input space.

Local Intrinsic Dimensionality (LID) [MLW+18]. The approach aims to explore the subspace surrounding adversarial examples. The *local intrinsic dimensionality* directly measures the expansion rate of the local distance distribution from a reference point to its nearest neighbors. The idea is that the expansion of the adversarial subspace is higher than that of the normal data subspace and hence the metric can be used to detect whether the input sample is adversarial or not. Differently from the methods before, LID features of the input samples are extracted at each output layer of the network of the pre-trained classifier.

Unsupervised methods

Methods falling under this category only rely on the features of natural samples that can be extracted using different techniques.

Feature Squeezing (FS) [XEQ18]. In this approach, the model's prediction on the original sample is compared to its prediction on the sample after *squeezing* (i.e., reducing the the color depth of images, and using smoothing to reduce the variation among the pixels). If the original and squeezed inputs produce substantially different outputs, the input is likely to be adversarial.

MagNet [MC17]. As in [FCSG17], the initial assumption is that the that adversarial examples lie off the true data manifold. A detector consists of an autoencoder that reconstructs input samples on the original manifold from input samples. The sample is classified as an adversary if the reconstruction error between the reconstructed input and the original exceeds a certain threshold. An additional autoencoder is used to improve performance, which calculates the Jensen's divergence between the original and reconstructed conditional distributions.

Part I

Misclassification Detection

CHAPTER 3

DOCTOR: A Simple Method for Detecting Misclassification Errors

In this chapter we tackle the problem of identifying whether the prediction of a classifier should (or should not) be trusted.

From the theoretical point of view, we derive the trade-off between two types of error probabilities: Type-I, that refers to the rejection of the classification for an input that would be correctly classified, and Type-II, that refers to the acceptance of the classification for an input that would not be correctly classified (Proposition 1). The characterization of the optimal discriminator in Eq. (3.7) allows us to devise a feasible implementation of it, based on the softmax probability (Proposition 2).

From the algorithmic point of view, inspired by our theoretical analysis, we propose DOCTOR a new discriminator (Definition 2), which yields a simple and flexible framework to detect whether a decision made by a model is likely to be correct or not. We distinguish two scenarios under wich DOCTOR can be deployed: Totally Black Box (TBB) where only the soft-predictions are available, hence gradient-propagation to perform input pre-processing is not allowed, and Partially Black Box (PBB) where we further allow method-specific inputs perturbations.

From the experimental point of view, we show that DOCTOR outperforms comparable state-of-the-art methods (e.g., ODIN [LLS18], softmax response [GE17] and Mahalanobis distance [LLLS18b]) on datasets including both in-distribution and out-of-distribution samples, and different architectures with various expressibilities, under both TBB and PBB. A key ingredient of DOCTOR is to fully

exploit all available information contained in the soft-probabilities of the predictions (not only their maximum).

3.1 The Optimal Discriminator

3.1.1 Statistical model for detection

Given a data sample $\mathbf{x} \in \mathcal{X}$ and an unobserved random label $y \in \mathcal{Y}$ drawn from the unknown distribution p_{XY}, we wish to predict the realization of the unobserved error variable $E \stackrel{\text{def}}{=} \mathbb{1}[Y \neq g_\theta(\mathbf{X})]$. To this end, we will model the data distribution as a mixture pdfs,

$$p_{XY}(\mathbf{x}, y) \equiv p_E(1)p_{XY|E}(\mathbf{x}, y|1) + p_E(0)p_{XY|E}(\mathbf{x}, y|0),$$

where $p_{XY|E}(\mathbf{x}, y|1)$ denotes the pdf truncated to the error event $\{E = 1\}$ (i.e., the hard decision fails) and $p_{XY|E}(\mathbf{x}, y|0)$ is the pdf truncated to the success event $\{E = 0\}$ (i.e., the hard decision succeeds). By taking the marginal of p_{XY} over the labels, we obtain: $p_X(\mathbf{x}) = p_E(1)p_{X|E}(\mathbf{x}|1) + p_E(0)p_{X|E}(\mathbf{x}|0)$. First, observe that the problem at hand is to infer E from $(\mathbf{x}, p_{\widehat{Y}|X})$ since Y is not observed. Second, we further emphasize that in the present framework we assume that there are no available (extra) samples for training a discriminator to distinguish between $p_{X|E}(\mathbf{x}|0)$ and $p_{X|E}(\mathbf{x}|1)$. It is worth mentioning that a well-trained classifier would imply $p_E(1) \ll p_E(0)$, since in that case we should have very few classification errors. However, this also implies that it would be very unlikely to have enough samples available to train a good enough discriminator.

3.1.2 Performance metrics and optimal discriminator

We aim to distinguish between samples for which the predictions cannot be trusted and samples for which predictions should be trusted. We first state the optimal rejection region, given by Eq. (3.1), where we suppose the existence of an oracle who knows all the involved probability distributions.

Definition 1 (Most powerful discriminator)**.** *For any $0 < \gamma < \infty$, define the de-*

cision region:

$$\mathcal{A}(\gamma) \stackrel{\text{def}}{=} \left\{\mathbf{x} \in \mathcal{X} : p_{X|E}(\mathbf{x}|1) > \gamma \cdot p_{X|E}(\mathbf{x}|0)\right\}. \tag{3.1}$$

The most powerful (Oracle) discriminator at threshold γ is defined by setting $D(\mathbf{x}, \gamma) = 1$ for all $\mathbf{x} \in \mathcal{A}(\gamma)$ for which the prediction is rejected (i.e., $\widehat{E} = 1$) and otherwise $D(\mathbf{x}, \gamma) = 0$ for all $\mathbf{x} \notin \mathcal{A}(\gamma)$ for which the prediction is accepted.

In Proposition 1, we establish the characterization of the fundamental performance of the most powerful (Oracle) discriminator by providing a lower bound on the error achieved by any discriminator and show that this bound is achievable by setting $\gamma = 1$. Furthermore, we connect this result to the Bayesian error rate of this optimal discriminator.

Proposition 1 (Performance of the discriminator)**.** *For any given decision region $\mathcal{A} \subset \mathcal{X}$, let*

$$\epsilon_0(\mathcal{A}) \stackrel{\text{def}}{=} \int_{\mathcal{A}} p_{X|E}(\mathbf{x}|0)d\mathbf{x}, \qquad \textit{and} \qquad \epsilon_1(\mathcal{A}^c) \stackrel{\text{def}}{=} \int_{\mathcal{A}^c} p_{X|E}(\mathbf{x}|1)d\mathbf{x}\,, \tag{3.2}$$

be the Type-I (rejection of the class prediction of an input $\mathbf{x}$ that would be correctly classified) and Type-II (acceptance of the class prediction of an input $\mathbf{x}$ that would not be correctly classified) error probability, respectively. Then,

$$\epsilon_0(\mathcal{A}) + \epsilon_1(\mathcal{A}^c) \geqslant 1 - \left\|p_{X|E=1} - p_{X|E=0}\right\|_{\text{TV}} \tag{3.3}$$

$$= 1 - \frac{1}{2}\int_{\mathcal{X}} |p_{X|E=1}(\mathbf{x}) - p_{X|E=0}(\mathbf{x})|d\mathbf{x}. \tag{3.4}$$

Equality is achieved by choosing the optimal decision region $\mathcal{A}^\star \equiv \mathcal{A}(1)$ in Definition 1. If the hypotheses are equally distributed, the minimum Bayesian error satisfies:

$$2\Pr\left\{D(\mathbf{X}) \neq E(\mathbf{X})\right\} \geqslant 1 - \left\|p_{X|E=1} - p_{X|E=0}\right\|_{\text{TV}}. \tag{3.5}$$

Equality is achieved by using the optimal decision region.

Expressions Eq. (3.4) and Eq. (3.5) provide lower bounds for the total error of an arbitrary discriminator. The proof of this proposition is relegated to the Supplementary material (Appendix A.1). Using Bayes we can rewrite Eq. (3.1)

via the posteriors as:

$$\mathcal{A}(\gamma) = \left\{\mathbf{x} \in \mathcal{X} : p_{E|X}(1|\mathbf{x})p_E(0) > \gamma \cdot \left(1 - p_{E|X}(1|\mathbf{x})\right) p_E(1)\right\}. \tag{3.6}$$

From Eq. (3.6), it is easy to check that $p_{E|X}(1|\mathbf{x}) = 1 - p_{Y|X}\left(g_\theta(\mathbf{x})|\mathbf{x}\right) = \mathsf{Pe}(\mathbf{x})$, and hence, the decision region $\mathcal{A}(\gamma)$ can be reformulated as:

$$\mathcal{A}(\gamma') = \left\{\mathbf{x} \in \mathcal{X} : \frac{\mathsf{Pe}(\mathbf{x})}{1 - \mathsf{Pe}(\mathbf{x})} > \gamma'\right\} = \left\{\mathbf{x} \in \mathcal{X} : \mathsf{Pe}(\mathbf{x}) > \frac{\gamma'}{(\gamma' + 1)}\right\}, \tag{3.7}$$

where $\gamma' \stackrel{\text{def}}{=} \gamma \cdot \frac{p_E(1)}{p_E(0)}$ and $0 < \gamma' < \infty$. According to Eq. (3.7) and Proposition Proposition 1, the optimal discriminator is given by $D^\star(\mathbf{x}, \gamma') = 1$, whenever $\mathbf{x} \in \mathcal{A}(\gamma')$, and $D^\star(\mathbf{x}, \gamma') = 0$, otherwise. The main difficulty arises here since the error probability function of an input: $\mathbf{x} \mapsto \mathsf{Pe}(\mathbf{x})$ is not known and in general cannot be learned from training samples.

3.2 The Proposed Discriminator: DOCTOR

3.2.1 DOCTOR discriminator

We start by deriving an approximation to the unknown function $\mathbf{x} \mapsto \mathsf{Pe}(\mathbf{x})$ which can be used to devise the decision region in expression (3.7). First, we state the following:

Proposition 2. *Let $\widehat{g}(\mathbf{x})$ be defined by*

$$1 - \widehat{g}(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{y \in \mathcal{Y}} p_{\widehat{Y}|X}(y|\mathbf{x};\theta) \Pr\left(\widehat{Y} \neq y|\mathbf{x}\right) = 1 - \sum_{y \in \mathcal{Y}} p^2_{\widehat{Y}|X}(y|\mathbf{x};\theta), \tag{3.8}$$

for each $\mathbf{x} \in \mathcal{X}$, which indicates the probability of incorrectly classifying a feature $\mathbf{x}$ if it was randomly labeled according to the model distribution $p_{\widehat{Y}|X}$ trained based on the dataset. Then,

$$(1 - \sqrt{\widehat{g}(\mathbf{x})}) - \Delta(\mathbf{x}) \leqslant \mathit{Pe}(\mathbf{x}) \leqslant (1 - \widehat{g}(\mathbf{x})) + \Delta(\mathbf{x}), \tag{3.9}$$

where $\Delta(\mathbf{x}) \stackrel{\text{def}}{=} 2\sqrt{2\, D_{\mathrm{KL}}\big(p_{Y|X}(\cdot|\mathbf{x}) \| p_{\widehat{Y}|X}(\cdot|\mathbf{x};\theta)\big)}$ and denotes the Kullback–Leibler (KL) divergence (further details are provided in Supplementary ma-

terial Appendix A.1.2).

3.2.2 Discussion

It is worth emphasizing that expressions in (3.9) provide bounds to the unknown function $\mathbf{x} \mapsto \mathsf{Pe}(\mathbf{x})$ using a known statistics $\mathbf{x} \mapsto 1 - \widehat{\mathsf{g}}(\mathbf{x})$, which is based on the soft-probability of the predictor. On the other hand, $0 \leqslant \widehat{\mathsf{g}}(\mathbf{x}) \leqslant \sqrt{\widehat{\mathsf{g}}(\mathbf{x})} \leqslant 1$, for all $\mathbf{x} \in \mathcal{X}$, which simply follows using the subadditive of the function $t \mapsto \sqrt{t}$ and the definition of $\widehat{\mathsf{g}}(\mathbf{x})$. By Markov's inequality,

$$\Pr\big(\Delta(\mathbf{X}) \geqslant \varepsilon(\eta)\big) \leqslant \eta \quad \text{with} \quad \varepsilon(\eta) = 2\sqrt{2\mathbb{E}_{\mathbf{X}Y}\big[-\log p_{\widehat{Y}|X}(Y|\mathbf{X};\theta)\big]/\eta}, \tag{3.10}$$

for any $\eta > 0$, where $\mathbb{E}_{\mathbf{X}Y}\big[-\log p_{\widehat{Y}|X}(Y|\mathbf{X};\theta)\big]$ in (3.10) is the cross-entropy risk. The latter is expected to be small provided that the model generalizes well. Thus, $\varepsilon(\eta)$ can be expected to be small for a desired confidence $\eta > 0$. Interestingly, (3.8) turns out to be related to the uncertainty of the classifier via the quadratic Rényi entropy [vEH14]: $-\log_2\big(\widehat{\mathsf{g}}(\mathbf{x})\big) = 2H_2(\widehat{Y}|\mathbf{x}) \leqslant 2H(\widehat{Y}|\mathbf{x})$, where the latter is the Shannon entropy, i.e., the self-uncertainty of the classifier.

3.2.3 From the theory to a practical discriminator

Our previous discussion suggests that $\widehat{\mathsf{Pe}}(\mathbf{x})$ in (2.2) may be a valuable candidate to approximate $\mathsf{Pe}(\mathbf{x})$ in the definition of the optimal discriminator (3.7). On the other hand, Proposition 2 suggests that $1-\widehat{\mathsf{g}}(\mathbf{x})$ can also be a valuable candidate yielding another discriminator. These discriminators, referred to as DOCTOR, are introduced below.

Definition 2 (DOCTOR)**.** *For any $0 < \gamma < \infty$ and $\mathbf{x} \in \mathcal{X}$, define the following discriminators:*

$$D_\alpha(\mathbf{x},\gamma) \stackrel{\text{def}}{=} \mathbb{1}\left[\widehat{g}(\mathbf{x}) > \gamma \cdot \widehat{g}(\mathbf{x})\right], \tag{3.11}$$

$$D_\beta(\mathbf{x},\gamma) \stackrel{\text{def}}{=} \mathbb{1}\left[\widehat{Pe}(\mathbf{x}) > \gamma \cdot (1 - \widehat{Pe}(\mathbf{x}))\right]. \tag{3.12}$$

Notice that because of Definition 2 and (3.8), $D_\alpha(\mathbf{x},\gamma) = \mathbb{1}[1 - \sum_{y\in\mathcal{Y}} \operatorname{softmax}^2(\mathbf{x})_y > \gamma \cdot \sum_{y\in\mathcal{Y}} \operatorname{softmax}^2(\mathbf{x})_y]$; similarly because of Definition 2 and eq. (2.2), $D_\beta(\mathbf{x},\gamma) = \mathbb{1}[1 - \max_{y\in\mathcal{Y}} \operatorname{softmax}(\mathbf{x})_y > \gamma \cdot \max_{y\in\mathcal{Y}} \operatorname{softmax}(\mathbf{x})_y]$.

The performance of these discriminators will be investigated and compared to state-of-the-art methods in the next section. In the Supplementary material (Appendix A.2), we illustrate how DOCTOR and the optimal discriminator (Definition 1) work on a synthetic data model that is a mixture of two spherical Gaussians with one component per class.

3.3 Evaluation

3.4 Experimental Results

In this section we present a collection of experimental results to investigate the effectiveness of DOCTOR, by applying it to several benchmark datasets. We provide publicly available code[1] to reproduce our results, and we give further details on the environment, the parameter setting and the experimental setup in the Supplementary material (Appendix A.3). We propose a comparison with state-of-the-art methods using similar information. Though we are not concerned with the OOD detection problem, we are confident it is appropriate to compare DOCTOR to methods which use soft-probabilities or at most the output of the latent code, e.g., ODIN [LLS18], softmax response (SR) [GE17] and Mahalanobis distance (MHLNB) [LLLS18b]. Since we are focusing on misclassification detection, it is expected that OOD samples should be also detected as classification errors.

Totally Black Box (TBB) and Partially Black Box (PBB). We address two different scenarios with respect to the available information about the network. In the TBB only the output of the last layer of the network is available, hence gradient-propagation to perform input pre-processing is not allowed. In the PBB we allow method-specific inputs perturbations. When considering DOCTOR in PBB, for each testing sample $\mathbf{x}$, we calculate the pre-processed sample $\tilde{\mathbf{x}}$ by adding a small perturbation:

$$\tilde{\mathbf{x}}^{\alpha} = \mathbf{x} - \epsilon \times \text{sign}\left[-\nabla_{\mathbf{x}} \log\left(\frac{1-\widehat{\mathrm{g}}(\mathbf{x})}{\widehat{\mathrm{g}}(\mathbf{x})}\right)\right],$$
$$\tilde{\mathbf{x}}^{\beta} = \mathbf{x} - \epsilon \times \text{sign}\left[-\nabla_{\mathbf{x}} \log\left(\frac{\widehat{\mathrm{Pe}}(\mathbf{x})}{1-\widehat{\mathrm{Pe}}(\mathbf{x})}\right)\right].$$

[1] https://github.com/doctor-public-submission/DOCTOR/

We will write directly $\widetilde{\mathbf{x}}$ when it is clear from the context which input pre-processing we are referring to. In Supplementary material (Appendix A.3.2) we further analyze the equations above. When ODIN or MHLNB are used, we pre-process the inputs as in [LLS18] and in [LLLS18b], respectively.

3.4.1 Review of related methods

PBB. We compare DOCTOR (using input pre-processing and temperature scaling) with ODIN and MHLNB. ODIN [LLS18] comprises temperature scaling and input pre-processing via perturbation. Temperature scaling is applied to its scoring function, which has $f_i(\widetilde{\mathbf{x}})$ for the logit of the i-th class. Formally, given an input sample $\mathbf{x}$:

$$\text{SODIN}(\widetilde{\mathbf{x}}) = \max_{i=[1:C]} \frac{\exp(f_i(\widetilde{\mathbf{x}})/T)}{\sum_{j=1}^{C} \exp(f_j(\widetilde{\mathbf{x}})/T)},$$

$$\text{ODIN}(\widetilde{\mathbf{x}}; \delta, T, \epsilon) = \begin{cases} \text{out}, & \text{if SODIN}(\widetilde{\mathbf{x}}) \leqslant \delta \\ \text{in}, & \text{if SODIN}(\widetilde{\mathbf{x}}) > \delta, \end{cases}$$

where $\widetilde{\mathbf{x}}$ represents a magnitude ϵ perturbation of the original $\mathbf{x}$; T is the temperature scaling parameter; $\delta \in [0, 1]$ is the threshold value; *in* indicates the acceptance decision while *out* indicates the rejection decision. Notice, however, γ in DOCTOR and δ in ODIN, respectively, are defined over two different domains: if δ denotes a probability, γ is a ratio between probabilities. Although ODIN originally required tuning the hyper-parameter T with out-of-distribution data, it was also shown that a large value for T is generally desirable, suggesting that this gain is achieved at 1000. Anyway, in this framework, we notice an improvement of ODIN in performance for low values of T. Thus we report the best results obtained by ODIN considering the range of hyper-parameters values tested also for DOCTOR (cf. Section 3.4.3). ENERGY [LWOL20] comprises the denominator

of the softmax activation:

$$\mathrm{ES}(\mathbf{x}) = -T \cdot \log \sum_{j=1}^{C} \exp(f_j(\mathbf{x})/T),$$

$$\mathrm{ENERGY}(\mathbf{x}; \xi, T) = \begin{cases} \text{out}, & \text{if } -\mathrm{ES}(\mathbf{x}) \leqslant \xi \\ \text{in}, & \text{if } -\mathrm{ES}(\mathbf{x}) > \xi, \end{cases}$$

where $\xi \in \mathbb{R}$ is the threshold value. Unlike all the methods considered in this paper, MHLNB [LLLS18b] requires the knowledge of the training set $\mathcal{D}_n$ which the pre-trained network was trained on to compute its *empirical class mean* $\widehat{\mu}_c$ for each class c and its *empirical covariance* $\widehat{\Sigma}$:

$$\widehat{\mu}_c = \frac{1}{n_c} \sum_{i:\, y_i = c} f(\widetilde{\mathbf{x}}_i); \;\; \widehat{\Sigma} = \frac{1}{n} \sum_{c \in \mathcal{Y}} \sum_{\forall i:\, y_i = c} (f(\widetilde{\mathbf{x}}_i) - \widehat{\mu}_c)(f(\widetilde{\mathbf{x}}_i) - \widehat{\mu}_c)^\top,$$

where n_c denotes the number of training samples with label c and $f(\widetilde{\mathbf{x}})$ the logits vector. As MHLNB directly uses the vector of logits, it does not comprise temperature scaling. Given an input sample $\mathbf{x}$:

$$\mathrm{M}(\widetilde{\mathbf{x}}) = \max_{c \in \mathcal{Y}} \; -(f(\widetilde{\mathbf{x}}) - \widehat{\mu}_c)^\top \widehat{\Sigma}^{-1} (f(\widetilde{\mathbf{x}}) - \widehat{\mu}_c),$$

$$\mathrm{MHLNB}(\widetilde{\mathbf{x}}; \zeta, \epsilon) = \begin{cases} \text{out}, & \text{if } \mathrm{M}(\widetilde{\mathbf{x}}) > \zeta \\ \text{in}, & \text{if } \mathrm{M}(\widetilde{\mathbf{x}}) \leqslant \zeta, \end{cases}$$

as mentioned above, $\widetilde{\mathbf{x}}$ represents a magnitude ϵ perturbation of the original $\mathbf{x}$; $\zeta \in \mathbb{R}_+$ is the threshold value; *in* indicates the acceptance decision while *out* indicates the rejection decision.

TBB. We compare DOCTOR (without input pre-processing and temperature scaling) with MHLNB (without input pre-processing and with the softmax output layer in place of the logits) and SR. Although both DOCTOR and SR have access to the softmax output of the predictor, a fundamental difference is that, while the former utilizes the softmax output in its entirety, the latter only uses the maximum value, therefore discarding potentially useful information. As it will be shown, this leads to better results for DOCTOR on several datasets (see Table 3.1). We emphasize that, by setting $T = 1$ and $\epsilon = 0$, ODIN reduces to softmax

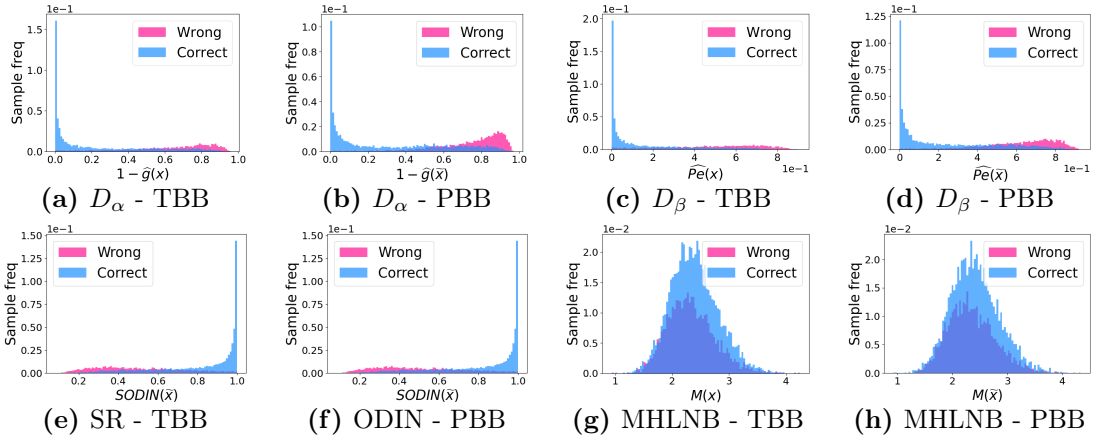
response [GE17] since $\text{SR}(\mathbf{x}) \equiv \text{SODIN}(\mathbf{x})$.

3.4.2 Detection of misclassification errors, experimental setup and evaluation metrics

Before digging into the detailed discussion of our numerical results, we present an empirical analysis of the behavior of DOCTOR, ODIN, SR and MHLNB when faced with the task of choosing whether to accept or reject the prediction of a given classifier for a certain sample. In Figure 3.1, we propose a graphical interpretation of the discrimination performance, considering the labeled samples in the dataset TinyImageNet and the ResNet network as the classifier. We separate correctly and incorrectly classified samples according to their true labels in blue and in red, respectively. We remind that the label information is *not* necessary for the discriminators to define acceptance and rejection regions. Then, for each sample we compute the corresponding discriminators' output. These values are binned and reported on the horizontal axis of Figure 3.1a and Figure 3.1b for D_α, Figure 3.1c and Figure 3.1d for D_β, Figure 3.1e for SR, Figure 3.1f for ODIN, Figure 3.1g and Figure 3.1h for MHLNB. In each each plot, and according to the corresponding discriminator, the bins' heights represent the frequency of the samples whose value falls within that bin. The intuition is that, if moving along the horizontal axis it is possible to pick a threshold value such that, w.r.t. this value, blue bars are on one side of the plot and red bars on the other, this threshold would correspond to the optimal discriminator, i.e. the discriminator that chooses the optimal acceptance and rejection regions.

In Figure 3.1g through Figure 3.1h, we observe that, for MHLNB, no matter how well we choose the threshold value, it is hard to fully separate red and blue bars both in TBB and PBB, i.e. the discriminator fails at defining acceptance and rejection regions so that all the hits can be assigned to the first one and all the mis-classification to the second one. The samples distribution for SR and ODIN in Figure 3.1e and Figure 3.1f, respectively, does not look significantly different from the one related to D_α and D_β in TBB (Equation (3.11)). However, the discrimination between samples becomes evident in PBB. This is shown in Figure 3.1d for D_β (eq. (3.11)) and even more in Figure 3.1b for D_α (eq. (3.11)) where, quite clearly, rightly classified samples are clustered on the left-end side of

the plot and incorrectly classified samples tend to cluster on the right-end side. This intuition is supported by the results in Table 3.1.



(a) D_α - TBB **(b)** D_α - PBB **(c)** D_β - TBB **(d)** D_β - PBB

(e) SR - TBB **(f)** ODIN - PBB **(g)** MHLNB - TBB **(h)** MHLNB - PBB

Figure 3.1: DOCTOR, ODIN, SR and MHLNB to split data samples in TinyImageNet both under TBB and PBB. (a) - (b) show the results for expressions (2.2); (c) - (d) show the results for (3.8); (e) shows the results for SR; (f) shows the results for ODIN; (g) - (h) show the results for MHLNB. Histograms for wrongly classified samples (red) and correctly classified samples (blue).

Datasets and pre-trained networks. We run experiments on both image and textual datasets. We use CIFAR10 and CIFAR100 [Kri09], TinyImageNet [JWX17] and SVHN [NWC+11] as image datasets; IMDb [MDP+11], AmazonFashion and AmazonSoftware [NLM19] as textual datasets. Note that, for all the aforementioned datasets, we consider only the test set since we rely on pre-trained models. Along the same lines of [LLS18], we use the pre-trained DenseNet models [HLW16] for CIFAR10, CIFAR100 and SVHN. In addition, we use a pre-trained ResNet model [HZRS16] for TinyImageNet, and BERT [DCLT19, WDS+20] for the Amazon datasets and IMDb. The accuracy achieved by the aforementioned networks on the test sets is showed in Table 3.1. According to the invariant properties of the discriminator (see Def. 2) with respect to the soft-probability of the underlying model, permutations of the posterior probabilities vector, due different initialization of the models before the training, do not change the output of Eq. (3.7), as it is a sum of squared values of the softmax probabilities. This variety of models/datasets characterizes the performance of the proposed method in scenarios with different accuracy levels.

Evaluation metrics. We will evaluate the performance according to Propo-

sition (1) via the empirical estimates of Type-I and Type-II errors in expressions (3.2). Throughout this section, when the model's decision for a sample is correct (hit) but is rejected by the discriminator, we refer to such event as *false rejection*; when the model's decision for a sample is not correct (miss) and is rejected by the discriminator, we refer to such event as *true rejection*. Similarly, we refer to a *false acceptance* when a miss is not rejected and to a *true acceptance* when a hit is not rejected. More specifically, let $\mathcal{T}_m = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\} \sim p_{XY}$ be the *testing set*, where $\mathbf{x}_i \in \mathcal{X}$ is the input sample, $y_i \in \{1, \ldots, C\}$ is the true class of $\mathbf{x}_i$, and m denotes the size of the testing set. With $j \in \{\alpha, \beta\}$:

$$\mathcal{FR}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y = g_\theta(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 1\}, \tag{3.13}$$

$$\mathcal{TR}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y \neq g_\theta(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 1\}, \tag{3.14}$$

$$\mathcal{FA}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y \neq g_\theta(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 0\}, \tag{3.15}$$

$$\mathcal{TA}_j(\gamma) = \{(\mathbf{x}, y) \in \mathcal{T}_m : y = g_\theta(\mathbf{x}),\ D_j(\mathbf{x}, \gamma) = 0\}\,. \tag{3.16}$$

We measure the performance of the test in terms of:

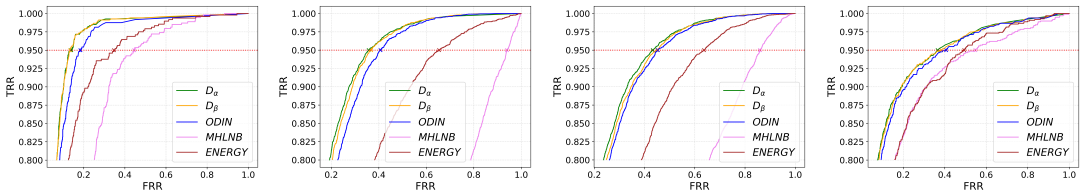
- **FRR** versus **TRR**. The false rejection rate (FRR) represents the probability that a hit is rejected, while the true rejection rate (TRR) is the probability that a miss is rejected.
- **AUROC**. The area under the *Receiver Operating Characteristic curve* (ROC) [DG06a] depicts the relationship between TRR and FRR. The perfect detector corresponds to a score of 100%.
- **FRR at 95 % TRR.** This is the probability that a hit is rejected when the TRR is at 95 %.

3.4.3 Experimental results: comparison between different discriminators

DOCTOR: comparison between D_α and D_β. We compare the discriminators D_α and D_β introduced in (3.11) to show how the AUROCs for CIFAR10, CIFAR100, TinyImageNet and SVHN change when varying the parameters T and ϵ. It is observed that D_α is less sensitive to the selection of T: for all the datasets, D_α outperforms D_β achieving the best AUROCs by setting $T = 1$. Contrary to

D_α, D_β is more sensitive to the value selected for T in the sense that small changes may result in very different values for the measured AUROCs (cf. Appendix A.3.4). In contrast,the best results are obtained for the same epsilon values of D_α and D_β across all the datasets.

Comparison in TBB. We compare DOCTOR with MHLNB (without input pre-processing and with the softmax output in place of the logits) and SR. It is worth to emphasize that D_α does not coincide (in general) with SR since the former consists in the sum of squared values of all probabilities involved in the softmax. To complete the comparison, we include the results for both methods in Table 3.1.



(a) CIFAR10 - PBB **(b)** CIFAR100 - PBB **(c)** Tiny - PBB **(d)** SVHN - PBB

Figure 3.2: ROC curves. Comparison between D_α ($T_\alpha = 1$ and $\epsilon_\alpha = 0.00035$), D_β ($T_\beta = 1.5$ and $\epsilon_\alpha = 0.00035$), ODIN ($T_{\text{ODIN}} = 1.3$ and $\epsilon_{\text{ODIN}} = 0$), MHLNB ($T_{\text{MHLNB}} = 1$ and $\epsilon_{\text{MHLNB}} = 0.0002$) and ENERGY ($T_{\text{ENERGY}} = 1$ and $\epsilon_{\text{ENERGY}} = 0$). Red dashed lines mark the 95% threshold of TRR.

Comparison in PBB. We compare DOCTOR with ODIN, MHLNB and ENERGY. We keep the same parameter setting for all the methods. In the case of DOCTOR and ODIN where temperature scaling is allowed, we test, for each dataset, 24 different values of ϵ for each of the 11 different values of T, see (Appendix A.3.4) for the set of ranges. In the case of MHLNB, which directly uses the logits, we keep $T = 1$ and we vary ϵ for each dataset. In the case of ENERGY, where no perturbation is allowed, we keep $\epsilon = 0$ and we maintain $T = 1$ (as in [LWOL20]). According to our framework, no validation samples are available; consequently, in order to be consistent across the datasets, we only report the experimental settings and values for which, on average, we obtain favorable results for all the considered domains (cf. Figure 3.2). In order to be fair, we update ODIN's parameters from those in [LLS18] to new values which are more suitable to the task at hand (cf. plots in Appendix A.3.4).

DOCTOR's performance compared to ODIN's, MHLNB's and ENERGY's,

are collected in Table 3.1 and in Figure 3.2. The results in the table show that noise further improves the performance of DOCTOR (cf. PBB) up to 1% over our previous experiments without noise (cf. TBB) in terms of AUROC. The improvement is even more significant in terms of FRR at 95% TRR: *a* 4% *decrease is obtained in terms of predictions incorrectly rejected for* DOCTOR *when passing from TBB to PBB*. Note that only the softmax output is available when we consider the pre-trained models for AmazonFashion, AmazonSoftware and IMDb datasets; therefore, we cannot access any internal layer and test DOCTOR for values of T which differ from the default value $T = 1$. Consequently, temperature scaling and input pre-processing cannot be applied in these cases and thus these datasets cannot be tested in PBB. Moreover, even in TBB, these datasets cannot be tested through MHLNB and ENERGY since the dataset on which the network was trained is not available. We provide simulations on how the range of interval for the different thresholds can affect the results in Appendix A.3.3.

Misclassification detection in presence of OOD samples. We evaluate DOCTOR's performance in misclassifcation detection considering a mixture of both in (DATASET-IN) and out-of-distribution (OOD) samples (DATASET-OUT), i.e. input samples for which the decision should not be trusted. The results are compared with ODIN. We test the two methods when one sample to reject out of five (♣), three (♢) or two (♠) is OOD. The details of the simulations, the considered dataset, and the complete experimental results are relegated Appendix A.3.4. In Table 3.2 we report an extract of the results for the PBB scenario in terms of *mean / standard deviation*: DOCTOR achieves, and most of the time outperforms ODIN's performance. We emphasize that, even though DOCTOR is not tuned for the OOD detection problem, it represents the best choice for deciding whether to accept or reject the prediction of the classifier also on mixed data scenarios where the percentage of OOD samples, as long as it is not dominant, can sensitively vary.

Table 3.1: Overall results for misclassification detection. For all methods, in TBB, we set $T = 1$ and $\epsilon = 0$; in PBB we set : $\epsilon_\alpha = \epsilon_\beta = 0.00035$, $T_\alpha = 1$, $T_\beta = 1.5$, $\epsilon_{\text{ODIN}} = 0$ and $T_{\text{ODIN}} = 1.3$, $\epsilon_{\text{MHLNB}} = 0.0002$ and $T_{\text{MHLNB}} = 1$, $\epsilon_{\text{ENERGY}} = 0$ and $T_{\text{ENERGY}} = 1$. In TBB, ODIN and SR coincide ($T = 1$ and $\epsilon = 0$).

DATASET	METHOD	AUROC↑%		FRR$_{\downarrow 95\%}$%	
		TBB	PBB	TBB	PBB
CIFAR10 Acc. 95%	D_α	**94**	**95.2**	**17.9**	13.9
	D_β	68.5	94.8	18.6	**13.4**
	ODIN	93.8	94.2	18.2	18.4
	SR	93.8	-	18.2	-
	MHLNB	92.2	84.4	30.8	44.6
	ENERGY	-	91.1	-	34.7
CIFAR100 Acc. 78%	D_α	**87**	**88.2**	40.6	**35.7**
	D_β	84.2	87.4	40.6	36.7
	ODIN	86.9	87.1	40.5	40.7
	SR	86.9	-	**40.5**	-
	MHLNB	82.6	50	66.7	94
	ENERGY	-	78.7	-	65.4
TINY IMAGENET Acc. 63%	D_α	**84.9**	**86.1**	**45.8**	**43.3**
	D_β	**84.9**	85.3	**45.8**	45.1
	ODIN	84.9	84.9	45.8	45.3
	SR	**84.9**	-	**45.8**	-
	MHLNB	78.4	59	82.3	86
	ENERGY	-	78.2	-	63.7

DATASET	METHOD	AUROC↑%		FRR$_{\downarrow 95\%}$%	
		TBB	PBB	TBB	PBB
SVHN Acc. 96%	D_α	**92.3**	**93**	**38.6**	**36.6**
	D_β	92.2	92.8	39.7	38.4
	ODIN	92.3	92.3	38.6	40.7
	SR	**92.3**	-	**38.6**	-
	MHLNB	87.3	88	85.8	54.7
	ENERGY	-	88.9	-	49.4
AMAZON FASHION Acc. 85%	D_α	**89.7**	-	27.1	-
	D_β	**89.7**	-	**26.3**	-
	SR	87.4	-	50.1	-
AMAZON SOFTWARE Acc. 73%	D_α	**68.8**	-	**73.2**	-
	D_β	**68.8**	-	**73.2**	-
	SR	67.3	-	86.6	-
IMDB Acc. 90%	D_α	**84.4**	-	**54.2**	-
	D_β	**84.4**	-	54.4	-
	SR	83.7	-	61.7	-

Table 3.2: Misclassification detection in presence of OOD samples. Same parameter setting as in Table 3.1 (PBB) for D_α, D_β, ODIN, ENERGY; as in [LLS18] for ODIN$_{\text{OOD}}$ and as in [LLLS18b] for MHLNB$_{\text{WB}}$. Results presented in terms of *mean / standard deviation*.

DATASET-IN	DATASET-OUT	AUROC↑%						FRR$_{\downarrow 95\%}$%					
		D_α	D_β	ODIN	ODIN$_{\text{OOD}}$	ENERGY	MHLNB$_{\text{WB}}$	D_α	D_β	ODIN	ODIN$_{\text{OOD}}$	ENERGY	MHLNB$_{\text{WB}}$
CIFAR10 ♣	iSUN	**95.4** / 0.1	95.1 / 0.1	94.6 / 0.1	89.6 / 0	92.4 / 0.1	54.5 / 0.1	14 / 0.5	**13.5** / 0.4	17.2 / 0.3	38.9 / 0	32.2 / 0.1	92 / 0.1
	TINY (RES)	**95.2** / 0.1	94.9 / 0	94.6 / 0.1	89.6 / 0	92.3 / 0.1	56.2 / 0	**14** / 0.4	**14** / 0.5	17.8 / 0.4	38.9 / 0	32.2 / 0.1	90.3 / 0.2
CIFAR10 ♢	iSUN	**95.5** / 0.1	95.3 / 0.1	94.9 / 0.1	91.5 / 0	92.9 / 0	54.5 / 0.1	14.4 / 0.6	**13.4** / 0.2	16.8 / 0.5	34/ 0.1	27 / 1	92 / 0.2
	TINY (RES)	**95.4** / 0.1	95 / 0.1	94.8 / 0.1	91.4 / 0	92.8 / 0	56.2 / 0.1	15 / 0.1	**14.8** / 0.7	17 / 0.5	34.5 / 0.9	28.8 / 1.9	90 / 0.3
CIFAR10 ♠	iSUN	**95.6** / 0.1	**95.6** / 0	95.4 / 0	93.5 / 0	93.6 / 0.1	54.6 / 0	15.1 / 0.1	**13.6** / 0.5	16.1 / 0.2	30.6 / 0.4	25.1 / 0.2	92 / 0.2
	TINY (RES)	**95.5** / 0.1	95.2 / 0.1	95.1 / 0.1	93.2	93.5 / 0	56.2 / 0.2	**14.7** / 0.3	14.8 / 0.5	17.1 / 0.4	31/ 0	25.6 / 0.3	90.2 / 0.1

3.5 Final remarks

Related work

Recent works have shown that the accuracy of a classifier and its ability to output soft-predictions that represent the estimate of the true posterior can be totally disjointed [GPSW17, KE16, KL15]. Furthermore, models often tend to be over-

confident about their decision even when their predictions fail [HAB19, KHH20]. This motivates a novel research area that strives to develop methods to assess when decisions made by classifiers should or should not be trusted. Although the detection of OOD samples is a different (domain) problem, it is naturally expected that samples from a distribution that is significantly different from the training one cannot be correctly classified. In [LLS18], the authors propose a method that increases the peakiness of the softmax output by perturbing the input samples and applying temperature scaling [GPSW17, HVD15, Pla00] to the classifier logits to detect in-distribution samples better. It is worth noticing that this method requires additional information on the internal structure of the latent code of the model. A very different approach [HSJK20, LLLS18b] tackles OOD detection by using the *Mahalanobis distance*. Although this approach appears to be more powerful, it also requires additional samples to learn the mean by class and the covariance matrix of the in-distribution. In [DT18], classifiers are trained to output calibrated confidence estimates that are used to perform OOD detection. A related line of research is concerned with the problem of *selective predictions* (aka *reject options*) in deep neural networks. The main motivation for selective prediction is reducing the error rate by abstaining from prediction when in doubt while keeping the number of correctly classified samples as high as possible [GKS21, GE17, GE19]. The idea is to combine classifiers with *rejection functions* by observing the classifiers' output, without supervision, to decide whether to accept or reject the classification outcome. In [GE17], the authors introduce *softmax response*, a rejection function which compares the maximum soft-probability to a pre-determined threshold to decide whether to accept or reject the class prediction given by the model.

Summary

We introduced a simple and effective method to detect misclassification errors, i.e., whether a classifier's prediction should or should not be trusted. We provided theoretical results on the optimal statistical model for misclassification detection, and we presented our empirical discriminator DOCTOR. Experiments on real (textual and visual) datasets–including OOD samples and comparison to state-of-the-art methods– demonstrate the effectiveness of our proposed meth-

ods. Whilst methods for ODD frameworks do not necessarily perform well in predicting misclassification errors, our result advances the state-of-the-art, and the main takeaway is that DOCTOR can be applied to both partially black-box (PBB) setups and totally black-box (TBB) ones. In the latter, information about the model's architecture may be undisclosed for security reason when dealing with sensitive data). DOCTOR uses all the information in the softmax output, which results in equal or better performance with respect to the other methods: the results in PBB, where we observe a reduction up to 4% in terms of predictions incorrectly rejected with respect to the ones in TBB are particularly promising. Moreover, DOCTOR does not require training data and, thanks to its flexibility, it can be easily deployed in real-world scenarios. Currently, DOCTOR does not exploit information across the layers yet. Only the soft-predictions are used. Besides, the most important obstacle is the calibration of the threshold (γ) between the desired fault rejection and acceptance rates, which would require additional validation samples. However, quite often, the cost of collecting data for this operation can be prohibitive, making it difficult or too expensive to perform such calibration. As future work, we shall combine DOCTOR with other related lines of research such as the extension to white-box incorporating additional information across the different latent codes of the model. Moreover, we shall investigate the possibility of combining the two proposed discriminators.

Part II

Multi-Armed Adversarial Attack Detection

CHAPTER 4

MEAD: A Multi-Armed Approach for Evaluation of Adversarial Examples Detectors

In this chapter we tackle the problem of adversarial attacks detection in a multi-armed framework, i.e. when the attacks at the evaluation time can be simultaneously crafted according to a variety of algorithms and objective loss functions.

We propose MEAD, a novel multi-armed evaluation framework for adversarial examples detectors involving several attackers to ensure that the detector is not overfitted to a particular attack strategy. The proposed metric is based on the following criterion. Each adversarial sample is correctly detected if and only if all the possible attacks on it are successfully detected. We show that this approach is less biased and yields a more effective metric than the one obtained by assuming only a single attack at evaluation time (cf. Section 4.3).

We make use of three new objective functions which, to the best of our knowledge, have never been used for the purpose of generating adversarial examples at testing time. These are *KL divergence*, *Gini Impurity* and *Fisher-Rao distance*. Moreover, we argue that each of them contributes to jointly creating competitive attacks that cannot be created by a single function (cf. Section 4.1).

We perform an extensive numerical evaluation of state-of-the-art (SOTA) and uncover their limitations, suggesting new research perspectives in this research line (cf. Section 4.4).

4.1 Generating adversarial examples according to different objectives

We recall the adversarial problem introduced in Section 2.2.1. Let us consider a natural sample $\mathbf{x} \in \mathcal{X}$ together with its true label $y \in \mathcal{Y}$. An attacker targets the model g_θ by crafting a sample $\mathbf{x}'_\ell \in \mathbb{R}^d$ according to an objective loss function $\ell(\mathbf{x}, \mathbf{x}'_\ell; \theta)$ which is denoted by ℓ, perturbation magnitude ε, and norm constraint L_p, $p \in \{1, 2, \infty\}$. The goal of the attack is to obtain an $\mathbf{x}'_\ell$ such that $g_\theta(\mathbf{x}'_\ell) \neq g_\theta(\mathbf{x})$, in order to force the target model to make a prediction error. The adversarial generation problem is commonly relaxed as follows

$$\mathbf{x}_\ell{}' \equiv \mathbf{x}_\ell{}'(\mathbf{x}) = \underset{\mathbf{x}_\ell{}' \in \mathbb{R}^d \,:\, \|\mathbf{x}_\ell{}' - \mathbf{x}\|_p < \varepsilon}{\operatorname{argmax}} \ell(\mathbf{x}, \mathbf{x}_\ell{}'; \theta), \tag{4.1}$$

where $\mathbf{x}'_\ell$ is updated iteration by iteration starting from an initial given value. The objective function ℓ traditionally used is the Adversarial Cross-Entropy (ACE) [SZS+14, MMS+18]:

$$\ell_{\mathrm{ACE}}(\mathbf{x}, \mathbf{x}_\ell{}'; \theta) = \mathbb{E}_{Y|\mathbf{x}}\big[-\log p_{\widehat{Y}|X}(Y|\mathbf{x}'_\ell; \theta)\big], \tag{4.2}$$

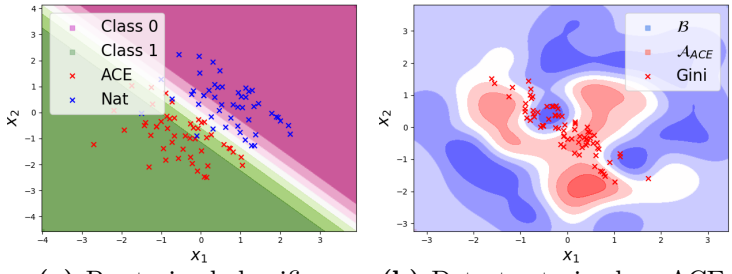
where the expectation is understood to be over the ground true conditional distribution of Y given $\mathbf{x}$. Inspired by recent development in the fields of robustness and misclassification detection [GRG+21, PMB+22, ZYJ+19], we include in our study recently proposed objective functions which generate diversified adversarial examples and that we briefly recall below.

- The Kullback-Leibler divergence (KL):

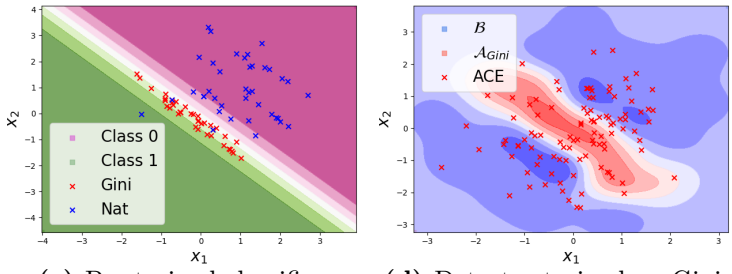
$$\ell_{\mathrm{KL}}\left(\mathbf{x}, \mathbf{x}_\ell{}'; \theta\right) = \mathbb{E}_{\widehat{Y}|\mathbf{x};\theta}\left[\log\left(\frac{p_{\widehat{Y}|X}(\widehat{Y}|\mathbf{x}; \theta)}{p_{\widehat{Y}|X}(\widehat{Y}|\mathbf{x}'_\ell; \theta)}\right)\right]. \tag{4.3}$$

- The Fisher-Rao objective (FR) [PMB+22]:

$$\ell_{\mathrm{FR}}(\mathbf{x}, \mathbf{x}_\ell{}'; \theta) = 2\arccos\left(\sum_{y \in \mathcal{Y}} \sqrt{p_{\widehat{Y}|X}(y|\mathbf{x}; \theta) p_{\widehat{Y}|X}(y|\mathbf{x}'_\ell; \theta)}\right). \tag{4.4}$$



(a) Pre-trained classifier **(b)** Detector trained on ACE



(c) Pre-trained classifier **(d)** Detector trained on Gini

Figure 4.1: ACE vs. Gini Impurity. Decision boundary for the binary classifier 4.1a-4.1c: the decision region for class 1 is green, the decision region of class 0 is pink. The natural testing samples belonging to class 0 are reported in blue, the corresponding adversarial examples crafted using ACE (4.1a) and Gini Impurity (4.1c) in red. Decision boundary of the detectors 4.1b-4.1d: $\mathcal{B}$, the decision region of the natural examples; $\mathcal{A}_\ell$, reported in red shades, the decision region of the adversarial examples when the detector is trained on data points crafted via $\ell \in \{\text{ACE}, \text{Gini}\}$ as objective. The darker shades stand for higher confidence. The red points represent the adversarial examples created with the opposite loss (respectively $\ell \in \{\text{Gini}, \text{ACE}\}$).

- The Gini Impurity score (Gini) [GRG+21]:

$$\ell_{\text{Gini}}(\cdot, \mathbf{x}_\ell{}'; \theta) = 1 - \sqrt{\sum_{y \in \mathcal{Y}} p^2_{\hat{Y}|X}(y|\mathbf{x}'_\ell; \theta)}. \tag{4.5}$$

  Interestingly, $\text{Gini}(\mathbf{x})$ corresponds to the function $1 - \sqrt{\hat{\text{g}}(\mathbf{x})}$ presented in Chapter 3.

4.2 A case study: ACE vs. Gini Impurity

In Figure 4.1 we provide insights on why we need to evaluate the detectors on attacks crafted through different objectives. We create a synthetic dataset that consists of 300 data points drawn from $\mathcal{N}_0 = \mathcal{N}(\mu_0, \sigma^2\mathbf{I})$ and 300 data points

drawn from $\mathcal{N}_1 = \mathcal{N}(\mu_1, \sigma^2\mathbf{I})$, where $\mu_0 = [1\ 1]$, $\mu_1 = [-1\ -1]$ and $\sigma = 1$. To each data point $\mathbf{x}$ is assigned true label 0 or 1 depending on whether $\mathbf{x} \sim \mathcal{N}_0$ or $\mathbf{x} \sim \mathcal{N}_1$, respectively. The data points have been split between the training set (70%) and the testing set (30%). We finally train a simple binary classifier with one single hidden layer and a learning rate of 0.01 for 20 epochs. We attack the classifier by generating adversarial examples with PGD under the L_∞-norm constraints with $\varepsilon = 1.2$ for the ACE attacks and $\varepsilon = 5$ for the Gini Impurity attacks to have a classification accuracy (classifier performance) of 50% on the corrupted data points. In Figure 4.1a-4.1c we plot the decision boundary of the binary classifier together with the adversarial and natural examples belonging to class 0. As can be seen, ACE creates points that lie in the opposite decision region with respect to the original points (Fig. 4.1a). Conversely, Gini Impurity tends to create new data points in the region of maximal uncertainty of the classifier (Fig. 4.1c). Consider the scenario where we train a simple Radial Basis Function (RBF) kernel SVM on a subset of the testing set of the natural points together with the attacked examples, generated with the ACE or the Gini Impurity score depending on the case (Fig. 4.1b-4.1d). We then test the detector on the data points originating with the opposite loss, Figure 4.1b and Figure 4.1d respectively. The decision region of the detector for natural examples is in blue, and the one for the adversarial examples is in red. The intensity of the color corresponds to the level of certainty of the detector. The detector's accuracy on natural and adversarial data points decreases from 71% to 62% when changing to the opposite loss in Fig. 4.1b, and from 87% to 63% in Fig. 4.1d. Hence, testing on samples crafted using a different loss in Eq. (2.4) means changing the attack and, consequently, evaluating detectors without considering this possibility leads to a more biased and unrealistic estimation of their performance. When the detector is trained on the adversarial examples created with both the losses, the accuracy is 79.8% when testing on Gini and 66.3% when testing on ACE, which is a better trade-off in adversarial detection performances.

The aforementioned losses will be included in the following section to design MEAD, our *multi-armed evaluation framework*, a new method to evaluate the performance of adversarial detection with low bias.

4.3 Evaluation with a Multi-Armed Attacker

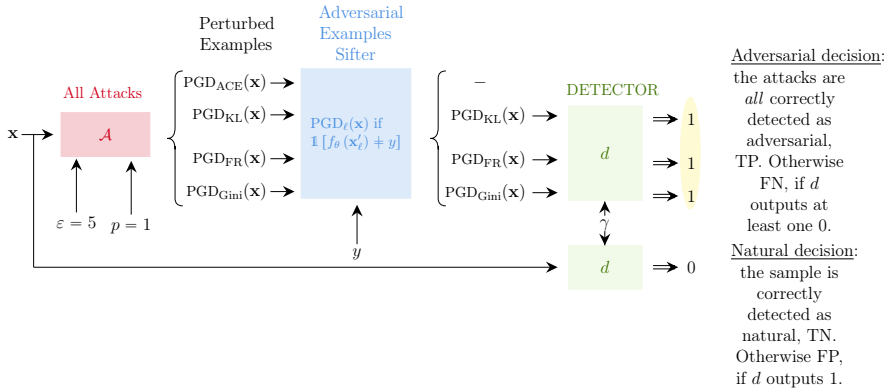


Figure 4.2: MEAD workflow: $\mathbf{x}$ is the natural example, $\varepsilon = 5$ is the perturbation magnitude, L_1 is the norm. From the set of all the possible existing attacks $\mathcal{A}$ we consider the ones using PGD. The sifter discards all the perturbed samples that do not fool the classifier g_θ. d is the detector.

The proposed evaluation framework, MEAD, consists in testing an adversarial examples detection method on a large collection of attacks grouped w.r.t. the L_p-norm and the maximal perturbation ε they consider. Each given natural input example is perturbed according to the collection of attacks. Note that, for every attack, a perturbed example is considered adversarial *if and only if* it fools the classifier. Otherwise, it is discarded and will not influence the evaluation. We then feed all the natural and successful adversarial examples to the detector and gather all the predictions. Finally, based on the detection decisions, we evaluate the detector according to a worst-case scenario:

i) Adversarial decision: for each natural example, we gather all the successful adversarial examples. If the detector detects *all* of them, then the perturbed sample is considered *correctly detected* (i.e., it is a true positive). However, if the detector misses at least one of them, the noisy sample is considered *undetected* (i.e., it is a false negative).

ii) Natural decision: for each natural sample, if the detector does not detect it, then the sample is considered *correctly non-detected* (i.e., it is a true negative); otherwise it is *incorrectly detected* (i.e., it is a false positive).

Specifically, let $\mathcal{D}_m = \{(\mathbf{x}_i, y_i)\}_{i=1}^m \sim p_{XY}$ be the testing set of size m, where $\mathbf{x}_i \in \mathcal{X}$ is the natural input sample and $y_i \in \mathcal{Y}$ is its true label. Let d: $\mathcal{X} \times \mathbb{R} \rightarrow$

$\{0,1\}$ be the detection mechanism and $a_\ell : \mathcal{X} \times \mathbb{R} \times \{1,2,\infty\} \to \mathcal{X}$ the attack strategy according to the objective function $\ell \in \mathcal{L}$ within a selected collection of objectives $\mathcal{L}$ as described in Section 4.1. For every considered L_p-norm, $p \in \{1,2,\infty\}$, maximal perturbation $\varepsilon \in \mathbb{R}$, and every threshold $\gamma \in \mathbb{R}$[1]:

$$Tp_{\varepsilon,p}(\gamma) = \Big\{(\mathbf{x},y) \in \mathcal{D}_m : \forall \ell \in \mathcal{L}\ \{g_\theta\big(a_\ell(\mathbf{x})\big) \neq y\} \ \wedge\ \{d\big(a_\ell(\mathbf{x}),\gamma\big) = 1\}\Big\} \tag{4.6}$$

$$FN_{\varepsilon,p}(\gamma) = \Big\{(\mathbf{x},y) \in \mathcal{D}_m : \exists \ell \in \mathcal{L}\ \{g_\theta\big(a_\ell(\mathbf{x})\big) \neq y\} \ \wedge\ \{d\big(a_\ell(\mathbf{x}),\gamma\big) = 0\}\Big\} \tag{4.7}$$

$$TN_{\varepsilon,p}(\gamma) = \{(\mathbf{x},y) \in \mathcal{D}_m : d(\mathbf{x},\gamma) = 0\} \tag{4.8}$$

$$Fp_{\varepsilon,p}(\gamma) = \{(\mathbf{x},y) \in \mathcal{D}_m : d(\mathbf{x},\gamma) = 1\}. \tag{4.9}$$

In Fig. 4.2 we provide a graphical interpretation of MEAD when the perturbation magnitude and the norm are fixed.

4.4 Experiments

In this section, we assess the effectiveness of the proposed evaluation framework, MEAD. The code is available at https://github.com/meadsubmission/MEAD.

4.4.1 Experimental setting

Evaluation metrics.

For each L_p-norm and each considered ε, we apply our multi-armed detection scheme. We gather the global result considering all the attacks and all the objectives. Moreover, we also report the results per objective. The performance is measured in terms of the AUROC↑% [DG06b] and in terms of $\text{FPR}\downarrow_{95\%}\%$. The first metric is the *Area Under the Receiver Operating Characteristic curve* and represents the ability of the detector to discriminate between adversarial and natural examples (higher is better). The second metric represents the percentage of natural examples detected as adversarial when 95 % of the adversarial examples are detected, i.e., FPR at 95 % TPR (lower is better).

[1]With an abuse of notation, $\forall \ell \in \mathcal{L}$ stands for all the considered attack mechanisms for specific values of ε, p within a collection of objectives $\mathcal{L}$.

CIFAR10	MEAD		ACE		KL		Gini		FR	
NSS	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
L_1 Average	**62.9**	**81.6**	67.4	75.7	67.1	76.0	67.8	78.2	67.6	75.6
L_2 Average	**64.0**	**82.0**	68.7	71.0	68.5	70.9	65.1	82.0	68.6	71.1
L_∞ Average	**71.9**	**62.0**	76.9	40.1	77.2	39.5	73.7	59.6	74.1	57.2
No norm	**88.5**	**38.8**	**88.5**	**38.8**	**88.5**	**38.8**	**88.5**	**38.8**	**88.5**	**38.8**
KD-BU	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
L_1 Average	**50.9**	**95.7**	70.0	88.6	70.0	88.4	74.3	92.3	69.8	88.4
L_2 Average	**59.0**	**94.1**	71.6	71.9	71.7	71.6	70.6	92.8	71.7	71.8
L_∞ Average	**36.8**	**96.9**	64.8	92.1	68.1	91.3	53.7	95.6	67.8	91.7
No norm	**65.4**	**94.2**	**65.4**	**94.2**	**65.4**	**94.2**	**65.4**	**94.2**	**65.4**	**94.2**
LID	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
L_1 Average	**50.8**	**95.4**	69.6	82.1	69.4	82.9	88.9	49.9	69.1	83.7
L_2 Average	**63.5**	**83.1**	73.7	70.1	73.4	70.7	82.5	61.3	73.2	71.3
L_∞ Average	**53.8**	**90.8**	75.7	56.8	79.9	57.6	71.3	79.7	82.0	51.4
No norm	**88.0**	**58.1**	**88.0**	**58.1**	**88.0**	**58.1**	**88.0**	**58.1**	**88.0**	**58.1**
FS	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
L_1 Average	75.4	64.8	92.8	25.1	92.9	24.9	**73.5**	**67.6**	92.9	24.6
L_2 Average	74.9	65.8	87.4	31.2	87.6	36.9	**73.7**	**67.2**	87.4	37.5
L_∞ Average	**52.7**	**81.1**	73.0	60.1	77.5	55.7	58.2	78.8	75.7	58.5
No norm	**62.7**	**82.5**	**62.7**	**82.5**	**62.7**	**82.5**	**62.7**	**82.5**	**62.7**	**82.5**
MagNet	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
L_1 Average	49.6	93.7	49.8	93.5	49.7	93.3	50.1	93.2	**49.1**	**93.8**
L_2 Average	50.9	93.1	52.3	89.6	52.3	89.4	**50.5**	**93.3**	51.8	91.4
L_∞ Average	**78.0**	**46.1**	79.2	44.6	80.2	44.1	79.2	44.6	80.0	44.6
No norm	**79.9**	**45.7**	**79.9**	**45.7**	**79.9**	**45.7**	**79.9**	**45.7**	**79.9**	**45.7**

Table 4.1: Average performance on CIFAR10 of all the detectors per objective and in MEAD. The worst results among all the settings is in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

MNIST	MEAD		ACE		KL		Gini		FR	
NSS	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
L_1 Average	**96.8**	**9.4**	97.0	8.2	97.1	8.6	97.4	7.0	97.1	8.1
L_2 Average	**90.3**	**26.5**	90.7	25.8	90.8	25.4	91.4	23.7	90.6	**26.5**
L_∞ Average	**88.7**	23.5	89.5	23.5	89.5	**23.6**	90.0	**23.6**	89.8	23.5
No norm	**87.1**	**57.8**	**87.1**	**57.8**	**87.1**	**57.8**	**87.1**	**57.8**	**87.1**	**57.8**
KD-BU	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
L_1 Average	**45.6**	**95.7**	59.9	93.0	59.3	93.1	61.4	92.7	58.9	93.3
L_2 Average	**50.3**	**94.8**	59.9	93.0	59.7	93.1	59.3	93.2	59.8	93.0
L_∞ Average	**34.1**	**96.7**	42.8	96.0	44.7	95.8	48.6	95.3	44.9	95.8
No norm	**76.0**	**88.2**	**76.0**	**88.2**	**76.0**	**88.2**	**76.0**	**88.2**	**76.0**	**88.2**
LID	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
L_1 Average	**79.9**	**54.9**	83.7	48.2	84.0	50.0	90.4	52.1	84.1	50.2
L_2 Average	**85.6**	**46.2**	87.4	44.1	87.0	45.1	87.6	44.4	86.1	45.4
L_∞ Average	**77.9**	**55.1**	83.3	46.3	83.6	47.8	88.7	38.8	83.0	49.5
No norm	**98.1**	**8.2**	**98.1**	**8.2**	**98.1**	**8.2**	**98.1**	**8.2**	**98.1**	**8.2**
FS	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
L_1 Average	**79.8**	**66.8**	83.4	57.6	83.5	57.1	83.2	53.0	83.4	57.4
L_2 Average	**73.5**	**69.0**	75.6	65.0	75.5	65.4	74.5	67.0	74.7	65.7
L_∞ Average	**76.4**	**63.5**	80.8	54.6	80.2	54.6	79.0	58.7	80.4	58.2
No norm	**61.5**	**85.9**	**61.5**	**85.9**	**61.5**	**85.9**	**61.5**	**85.9**	**61.5**	**85.9**
MagNet	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
L_1 Average	**98.1**	5.7	98.2	5.4	98.3	**5.6**	98.3	5.2	**98.1**	**5.6**
L_2 Average	90.0	28.7	90.6	27.6	90.8	27.8	90.6	**29.1**	**89.7**	28.1
L_∞ Average	98.5	10.3	98.5	10.3	**98.4**	**10.6**	98.5	10.5	98.5	10.4
No norm	**86.9**	**74.3**	**86.9**	**74.3**	**86.9**	**74.3**	**86.9**	**74.3**	**86.9**	**74.3**

Table 4.2: Average performance on MNIST of all the detectors per objective and in MEAD. The worst results among all the settings is in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

	Supervised Methods						Unsupervised Methods			
	NSS		*KD-BU*		*LID*		*FS*		*MagNet*	
	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
MNIST	90.7	**29.3**	51.5	93.9	85.4	41.1	72.8	71.3	**93.4**	29.8
CIFAR10	**71.8**	**66.1**	53.0	95.2	64.0	81.8	66.4	73.6	64.6	69.7

Table 4.3: Performances of each detection method under the MEAD framework on CIFAR10 and MNIST averaged over the norm-based constraint. The best results among all the methods is in **bold**; the ones per type of detection method (i.e. Supervised and Unsupervised) are underlined.

Datasets and classifiers.

We run the experiments on MNIST [LCB10] and CIFAR10 [Kri09]. The underlying classifiers are a simple CNN for MNIST, consisting of two blocks of two convolutional layers, a max-pooling layer, one fully-connected layer, one dropout layer, two fully-connected layers, and ResNet-18 for CIFAR10. The training procedures involve 100 epochs with Stochastic Gradient Descent (SGD) optimizer using a learning rate of 0.01 for the simple CNN and 0.1 for ResNet-18; a momentum of 0.9 and a weight decay of 10^{-5} for ResNet-18. Once trained, these networks are fixed and never modified again.

Grouping attacks.

We test the methods on the attacks presented in Section 2.2.3, and we present them based on the norm constraint used to construct the attacks.Under the L_1-norm fall PGD with ε in $\{5, 10, 15, 20, 25, 30, 40\}$. Under the L_2-norm fall PGD with ε in $\{0.125, 0.25, 0.3125, 0.5, 1, 1.5, 2\}$, CW with $\varepsilon = 0.01$, HOP with $\varepsilon = 0.1$, and DF which has no constraint on ε. Under the L_∞-norm fall FGSM, BIM and PGD with ε in $\{0.0315, 0.0625, 0.125, 0.25, 0.3125, 0.5\}$, CW with $\varepsilon = 0.3125$, and SA with $\varepsilon = 0.3125$ for MNIST and $\varepsilon = 0.125$ for CIFAR10. Finally, ST is not constrained by a norm or a maximum perturbation, as it is limited in maximum rotation (30 for CIFAR10 and 60 for MNIST) and translation (8 for CIFAR10 and 10 for MNIST).

Detection Methods.

We tested detection methods introduced in Section 2.2.4. In the supervised case, we train the detectors using adversarial examples created by perturbing the samples in the original training sets with PGD under L_∞-norm and $\varepsilon = 0.03125$. In

the unsupervised case, the detectors only need natural samples in the training sets. They are tested on all the previously mentioned attacks, generated on the testing sets.

4.4.2 Experimental results

In this section, we refer to *single-armed setting* when we consider the setup where the adversarial examples are generated w.r.t. one of the objectives in Section 4.1. We provide the average of the performances of all the detection methods on CIFAR10 in Table 4.1 and on MNIST in Table 4.2. The detailed tables for each detection method (i.e., *NSS*, *LID*, *KD-BU*, *MagNet*, and *FS*) and for each dataset (i.e., CIFAR10 and MNIST) are reported in Appendix B.1.

MEAD and the single-armed setting.

Table 4.1 shows a decrease in the performance of all the detectors when going from the single-armed setting to MEAD. *NSS* is the more robust among the supervised methods when passing from the single-armed setting to the proposed setting. Indeed, (cf Table 4.1), in terms of AUROC↑%, it registers a decrease of up 4.9 percentage points under the L_1-norm constraint, 4.7 under the L_2-norm constraint, and 5.3 under the L_∞-norm constraint. This can be explained by the fact that the network in *NSS* is trained on the natural scene statistics extracted from the trained samples differently from the other detectors. In particular, these statistical properties are altered by the presence of adversarial perturbations and hence are found to be a good candidate to determine if a sample is adversarial or not. By looking closely at the results for *NSS* in Table B.2, it comes out that it performs better when evaluated on L_∞ norm constraint. Indeed, in this case, the adversarial examples at testing time are similar to those used at training time. Not surprisingly, the performance decreases when evaluated on other kinds of attacks. Notice that, in the single-armed setting, all the supervised methods turn out to be much more inefficient than when presented in the original papers. Indeed, as already explained in Section 4.4.1, we train the detectors using adversarial examples created by perturbing the samples in the original training sets with PGD under L_∞-norm and $\varepsilon = 0.03125$, and then we test them on a variety of attacks. Hence, we do not train a different detector for each kind of attack seen

at testing time. On the other side, the unsupervised detector *MagNet* appears to be more robust than *FS* when changing from the single-armed setting to MEAD. Indeed, in terms of AUROC↑%, it loses at most 2.2 percentage points (L_∞ norm case). On average, *FS* is the unsupervised detector that achieves the best performance on CIFAR10, while *MagNet* is the one to achieve the best performance on MNIST.

Remark: Some single-armed setting results turn out to be worse than the corresponding results in MEAD (cf Table B.2-B.6 and Table B.8-B.12 in Appendix B.1). We provide here an explanation of this phenomenon. Given a natural input sample $\mathbf{x}$, let $\mathbf{x}_\ell$ denotes the perturbed version of $\mathbf{x}$ according to some fixed norm p, fixed perturbation magnitude ε and objective function ℓ between ACE, KL, Gini and FR. Suppose $g_\theta(\mathbf{x}_{\text{ACE}}) = y$, where y is the ground true label of $\mathbf{x}$, this means that $\mathbf{x}_{\text{ACE}}$ is a perturbed version of the natural example but not adversarial. Assume instead $g_\theta(\mathbf{x}_{\text{KL}}) \neq y$, $g_\theta(\mathbf{x}_{\text{Gini}}) \neq y$ and $g_\theta(\mathbf{x}_{\text{FR}}) \neq y$. If at testing time the detector is able to recognize all of them as being positive (i.e., adversarial), then under MEAD, $\mathbf{x}_{\text{KL}}, \mathbf{x}_{\text{Gini}}, \mathbf{x}_{\text{FR}}$ would be considered a *true positive*. This example, counting as a true positive under MEAD, would instead be discarded under the single-armed setting of ACE, as $\mathbf{x}_{\text{ACE}}$ is neither a clean example nor an adversarial one. Then, the larger amount of true positives in MEAD can potentially lead to an increase in the global AUROC↑%.

Effectiveness of the proposed objective functions.

In Table B.1 and Table B.7, relegated to the Appendix, we report the averaged number of successful adversarial examples under the multi-armed setting as well as the details per single-armed settings on CIFAR10 and MNIST, respectively. The attacks are most successful when the value of the constraint ε for every L_p-norm increases. Generating adversarial examples using the ACE for each attack scheme creates more harmful (adversarial) examples for the classifier than using any other objective. However, using either the Gini Impurity score, the Fisher-Rao objective, or the Kullback-Leibler divergence seems to create examples that are either equally or more difficult to be detected by the detection methods. For this purpose, we provide two examples. First, by looking at the results in Table B.4, we can deduce that *LID* finds it difficult to recognize the attacks

based on KL and FR objective functions but not the ones created through Gini. For example, with PGD1 and $\varepsilon = 40$, we register a decrease in AUROC↑% of 9.5 percentage points when going from the single-armed setting of Gini to the one of FR. Similarly, the decrease is 8.3 percentage points in the case of KL. This behavior is even more remarkable when we look at the results in terms of FPR$\downarrow_{95\%}$%: the gap between the best FPR$\downarrow_{95\%}$% values (obtained via Gini) and the worst (via FR) is 30.7 percentage points. On the other side, the situation is reversed if we look at the results in Table B.5 as *FS* turns out to be highly inefficient at recognizing adversarial examples generated via the Gini Impurity score. By considering the results associated to the highest value of ε for each norm, namely $\varepsilon = 40$ for L_1-norm; $\varepsilon = 2$ for L_2-norm; $\varepsilon = 0.5$ for L_∞-norm, the gap between best FPR$\downarrow_{95\%}$% values (obtained via KL divergence) and the worst (via Gini Impurity score), varies from a minimum of 41.7 (L_∞-norm) to a maximum of 64.4 (L_2-norm) percentage points. This example, in agreement with Section 4.2, testify on real data that testing the detectors without taking into consideration the possibility of creating attacks through different objective functions leads to a biased and unrealistic estimation of their performances.

Comparison between supervised and unsupervised detectors.

The unsupervised methods find it challenging to recognize attacks crafted using the Gini Impurity score. Indeed, according to Section 4.2, that objective function creates attacks on the decision boundary of the pre-trained classifier. Consequently, the unsupervised detectors can easily associate such input samples with the cluster of naturals. Supervised methods detect Adversarial Cross-Entropy loss-based attacks more and, therefore, more volatile when it comes to other types of loss-based attacks. Overall, by looking at the results in Table 4.3 on both the datasets, most of the supervised and unsupervised methods achieve comparable performances with the multi-armed framework, meaning that the current use of the knowledge about the specific attack is not general enough. The exception to this is *NSS*, which, as already explained, seems to be the most general detector.

On the effects of the norm and ε.

The detection methods recognize attacks with a large perturbation more easily than other attacks (cf Table B.2-B.6 and Table B.8-B.12). L_∞-norm attacks are less easily detectable than any other L_p-norm attack. Indeed, multiple attacks are tested simultaneously for a single ε under the L_∞ norm constraint. For example, in CIFAR10 with $\varepsilon = 0.3125$ and L_∞, PGD, FGSM, BIM, and CW are tested together, whereas, with any other norm constraint, only one typology of attack is examined. Indeed the more attack we consider for a given ε, the more likely at least one attack will remain undetected. Globally, under the L_∞-norm constraint, Gini Impurity score-based attacks are the least detected attacks. However, each method has different behaviors under L_1 and L_2. *NSS* is more sensitive to Kullback-Leibler divergence-based attacks while *MagNet* is more volatile to the Fisher-Rao distance-based attacks. As already pointed out, *FS* achieves inferior performance when evaluated against attacks crafted through the Gini Impurity objective, while the sensitivity of *LID* and *KD-BU* to a specific objective depends on the L_p-norm constraint.

4.5 Final remarks

We introduced MEAD a new framework to evaluate detection methods of adversarial examples. Contrary to what is generally assumed, the proposed setup ensures that the detector does not know the attacks at the testing time and is evaluated based on simultaneous attack strategies. Our experiments showed that the SOTA detectors for adversarial examples (both supervised and unsupervised) mostly fail when evaluated in MEAD with a remarkable deterioration in performance compared to single-armed settings. We enrich the proposed evaluation framework by involving three new objective functions to generate adversarial examples that create adversarial examples which can simultaneously fool the classifier while not being successfully identified by the investigated detectors. The poor performance of the current SOTA adversarial examples detectors should be seen as a challenge when developing novel methods. However, our evaluation framework assumes that the attackers do not know the detection method. As future work we plan to enrich the framework to a complete whitebox scenario.

CHAPTER 5

A Minimax Approach Against Multi-Armed Adversarial Attack Detection

In this chapter, we propose an aggregation framework to combine the expertise of different adversarial examples detectors and address the problem of simultaneous attack detection as highlighted in Chapter 4. This method can aggregate pre-trained detectors without the need for additional training.

From a theoretical perspective, we revisit the multi-armed attack detection problem formulated in Chapter 4 and formalize it as a minimax cross-entropy risk. Based on this formulation, we derive a surrogate loss function and use it to characterize our optimal soft-detector in Eq. (5.5), leading to our proposed solution.

Empirical evaluations of our proposed solution on popular datasets, such as CIFAR10 and SVHN, show that it leads to higher and more consistent performance compared to the state-of-the-art (SOTA) in the simultaneous attack setup, even when using simple detectors that individually perform worse than SOTA detectors, as demonstrated in Section 5.2.

5.1 Formalization of the Problem of Detecting multi-armed Adversarial Attacks

In this section, we begin by formalizing the problem of multi-armed attacks as proposed in Chapter 4. We then delve deeper into the topic of optimal detectors and demonstrate how to apply our proposed solution to practical use-cases.

5.1.1 Statistical model

Let $\mathcal{K}$ be the countable set of indexes corresponding to each possible attack, e.g., based on various attack algorithms and loss functions, as described in Section 2.2.1. Let $\mathcal{M} = \left\{P_{XZ}^{(k)} : k \in \mathcal{K}\right\}$ be the set of joint probability distributions on $\mathcal{X} \times \mathcal{Z}$ which are indexed with k, $\forall k \in \mathcal{K}$, where $\mathcal{X}$ is the input (feature) space and $\mathcal{Z} = \{0, 1\}$ indicates a binary space label for the adversarial example detection task. At the evaluation time, the attacker selects an arbitrary strategy $k \in \mathcal{K}$ and then samples an input according to $p_{X|Z}^{(k)}(\mathbf{x}|z = 1)$ which corresponds to the probability density function induced by the chosen attack k where $p_{X|Z}^{(k)}(\mathbf{x}|z = 0) = p_X(\mathbf{x})$ *almost surely* corresponds to the probability distribution of the natural samples. The learner is given a set of *soft-detectors* models:

$$\mathcal{Q} = \left\{q_{\hat{Z}|\mathbf{u}}^{(k)} : \mathcal{U} \mapsto [0, 1]^2\right\}_{k \in \mathcal{K}},$$

which have possibly been trained to detect attacks according to each strategy $k \in \mathcal{K}$, e.g., $q_{\hat{Z}|\mathbf{u}}^{(k)} \equiv p_{\hat{Z}|U}(z|\mathbf{u}; \psi_k)$ with parameters ψ_k and $\mathbf{u} \in \mathcal{U} = \{g_\theta^l(\mathbf{x}) \mid \mathbf{x} \in \mathbb{R}^d\}$ denotes the space of logits. The set of possible detectors $\mathcal{Q}$ is available to the defender. However, the specific attack chosen by the attacker at the test time is unknown. In the remainder of this section, we formally devise an optimal detector that exploits full knowledge of the set $\mathcal{Q}$.

5.1.2 A novel objective for detection under simultaneous attacks

Consider a fixed input sample $\mathbf{x_0}$ and let $\mathbf{u_0} = g_\theta^l(\mathbf{x_0})$. Clearly, the problem at hand consists in finding an optimal soft-detector $q_{\hat{Z}|\mathbf{u_0}}^\star$ that performs well simultaneously over all possible attacks in $\mathcal{K}$. This can be formalized as the solution

to the following minimax problem:

$$\mathcal{L}(\mathcal{Q}, \mathbf{x_0}) = \min_{q_{\hat{Z}|\mathbf{u_0}}} \max_{k \in \mathcal{K}} \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[-\log q_{\hat{Z}|\mathbf{u_0}} \right], \tag{5.1}$$

which requires to solve (5.1) for $\mathcal{Q}$ and for each given input sample $\mathbf{x_0}$. It is important to note that the minimization is performed over all (detectors) distributions $q_{\hat{Z}|\mathbf{u_0}}$, including elements that are not part of the set $\mathcal{Q}$.

That being said, the objective in Eq. (5.1) is not tractable computationally. To overcome this issue, we derive a surrogate (an upper bound) that can be computationally optimized. For any arbitrary choice of $q_{\hat{Z}|\mathbf{u_0}}$, we have

$$\max_{k \in \mathcal{K}} \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[-\log q_{\hat{Z}|\mathbf{u_0}} \right] \leqslant \underbrace{\max\nolimits_{k \in \mathcal{K}} \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[-\log q^{(k)}_{\hat{Z}|\mathbf{u_0}} \right]}_{=\text{constant term}} + \max_{k \in \mathcal{K}} \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[\log \left(\frac{q^{(k)}_{\hat{Z}|\mathbf{u_0}}}{q_{\hat{Z}|\mathbf{u_0}}} \right) \right]. \tag{5.2}$$

Proof of Eq. (5.2).

$$\begin{aligned} \max_{k \in \mathcal{K}} \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[-\log q_{\hat{Z}|\mathbf{u_0}} \right] &= \max_{k \in \mathcal{K}} \left[\mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[-\log q^{(k)}_{\hat{Z}|\mathbf{u_0}} \right] + \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[\log \left(\frac{q^{(k)}_{\hat{Z}|\mathbf{u_0}}}{q_{\hat{Z}|\mathbf{u_0}}} \right) \right] \right] \\ &\leqslant \max_{k \in \mathcal{K}} \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[-\log q^{(k)}_{\hat{Z}|\mathbf{u_0}} \right] + \max_{k \in \mathcal{K}} \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[\log \left(\frac{q^{(k)}_{\hat{Z}|\mathbf{u_0}}}{q_{\hat{Z}|\mathbf{u_0}}} \right) \right]. \end{aligned}$$

□

Observe that the first term in (5.2) of the upper bound is constant w.r.t. the choice of $q_{\hat{Z}|\mathbf{u_0}}$ and the second term is well-known as being equivalent to the *average worst-case regret* [BRY98]. This upper bound provides a surrogate to our intractable objective in (5.1) that can be minimized over all $q_{\hat{Z}|\mathbf{u_0}}$. We can formally state our problem as follows:

$$\tilde{\mathcal{L}}(\mathcal{Q}, \mathbf{x_0}) = \min_{q_{\hat{Z}|\mathbf{u_0}}} \max_{k \in \mathcal{K}} \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[\log \left(\frac{q^{(k)}_{\hat{Z}|\mathbf{u_0}}}{q_{\hat{Z}|\mathbf{u_0}}} \right) \right] = \min_{q_{\hat{Z}|\mathbf{u_0}}} \max_{P_\Omega} \mathbb{E}_\Omega \left[D_{\mathrm{KL}} \left(q^{(\Omega)}_{\hat{Z}|\mathbf{u_0}} \| q_{\hat{Z}|\mathbf{u_0}} \right) \right], \tag{5.3}$$

where the min is taken over all the possible distributions $q_{\hat{Z}|\mathbf{u_0}}$; and Ω is a discrete random variable with P_Ω denoting a generic probability distribution whose probabilities are $(\omega_1, \ldots, \omega_{|\mathcal{K}|})$, i.e., $P_\Omega(k) = \omega_k$; and $D_{\mathrm{KL}}(\cdot\|\cdot)$ is the Kullback–Leibler divergence, representing the expected value of regret of $q_{\hat{Z}|U}$ w.r.t. the worst-case distribution in $\mathcal{Q}$.

Proof of Eq. (5.3). The equality hold by noticing that

$$\max_{P_\Omega} \mathbb{E}_\Omega \left[D_{\mathrm{KL}} \left(q^{(\Omega)}_{\hat{Z}|\mathbf{u_0}} \| q_{\hat{Z}|\mathbf{u_0}} \right) \right] \leqslant \max_{k \in \mathcal{K}} \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[\log \left(\frac{q^{(k)}_{\hat{Z}|\mathbf{u_0}}}{q_{\hat{Z}|\mathbf{u_0}}} \right) \right],$$

and moreover,

$$\max_{k \in \mathcal{K}} \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[\log \left(\frac{q^{(k)}_{\hat{Z}|\mathbf{u_0}}}{q_{\hat{Z}|\mathbf{u_0}}} \right) \right] = \mathbb{E}_{\bar{\Omega}} \left[D_{\mathrm{KL}} \left(q^{(\bar{\Omega})}_{\hat{Z}|\mathbf{u_0}} \| q_{\hat{Z}|\mathbf{u_0}} \right) \right],$$

by choosing the random variable $\bar{\Omega}$ with uniform probability over the set of maximizers $\overline{\mathcal{K}} = \mathrm{argmax}_{k \in \mathcal{K}} \mathbb{E}_{q^{(k)}_{\hat{Z}|\mathbf{u_0}}} \left[\log \left(\frac{q^{(k)}_{\hat{Z}|\mathbf{u_0}}}{q_{\hat{Z}|\mathbf{u_0}}} \right) \right]$, zero otherwise. □

The convexity of the KL-divergence allows us to rewrite Eq. (5.3) as follows:

$$\min_{q_{\hat{Z}|\mathbf{u_0}}} \max_{P_\Omega} \mathbb{E}_\Omega \left[D_{\mathrm{KL}} \left(q^{(\Omega)}_{\hat{Z}|\mathbf{u_0}} \| q_{\hat{Z}|\mathbf{u_0}} \right) \right] = \max_{P_\Omega} \min_{\hat{q}_{\hat{Z}|\mathbf{u_0}}} \mathbb{E}_\Omega \left[D_{\mathrm{KL}} \left(q^{(\Omega)}_{\hat{Z}|\mathbf{u_0}} \| q_{\hat{Z}|\mathbf{u_0}} \right) \right]. \tag{5.4}$$

Proof of Eq. (5.4). We consider a zero-sum game with a concave-convex mapping defined on a product of convex sets. The sets of all probability distributions $q_{\hat{Z}|\mathbf{u_0}}$ and P_Ω are two nonempty convex sets, bounded and finite dimensional. On the other hand, $\left(P_\Omega, q_{\hat{Z}|\mathbf{u_0}}\right) \to \mathbb{E}_\Omega \left[D_{\mathrm{KL}} \left(q^{(\Omega)}_{\hat{Z}|\mathbf{u_0}} \| q_{\hat{Z}|\mathbf{u_0}} \right) \right]$ is a concave-convex mapping, i.e., $P_\Omega \to \mathbb{E}_\Omega \left[D_{\mathrm{KL}} \left(q^{(\Omega)}_{\hat{Z}|\mathbf{u_0}} \| q_{\hat{Z}|\mathbf{u_0}} \right) \right]$ is concave and $q_{\hat{Z}|\mathbf{u_0}} \to \mathbb{E}_\Omega \left[D_{\mathrm{KL}} \left(q^{(\Omega)}_{\hat{Z}|\mathbf{u_0}} \| q_{\hat{Z}|\mathbf{u_0}} \right) \right]$ is convex for every $\left(P_\Omega, q_{\hat{Z}|\mathbf{u_0}}\right)$. Then, by classical min-max theorem [vN28] we have that Eq. (5.4) holds. □

The solution to Eq. (5.4) provides the optimal distribution $P^\star_\Omega$, i.e. the col-

lection of weights $\{w_k^\star\}$, which leads to our soft-detector [BRY98]:

$$\widehat{q}^{\star}_{\widehat{Z}|\mathbf{u_0}} = \sum_{k\in\mathcal{K}} w_k^\star \cdot q^{(k)}_{\widehat{Z}|\mathbf{u_0}}, \quad \text{with} \quad P^\star_\Omega = \underset{\{\omega_k\}}{\operatorname{argmax}}\, I_{\mathbf{u_0}}(\Omega;\widehat{Z}), \tag{5.5}$$

where $I_{\mathbf{u_0}}(\cdot;\cdot)$ denotes the Shannon mutual information between the random variable Ω, distributed according to $\{\omega_k\}$, and the binary soft-prediction variable $\widehat{Z}$, distributed according to $q^{(k)}_{\widehat{Z}|\mathbf{u_0}}$ and conditioned on the particular test example $\mathbf{u_0}$.

Proof of Eq. (5.5). It is enough to show that

$$\min_{\widehat{q}_{\widehat{Z}|\mathbf{u_0}}} \mathbb{E}_\Omega\left[D_{\mathrm{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}} \| q_{\widehat{Z}|\mathbf{u_0}}\right)\right] = I_{\mathbf{u_0}}(\Omega;\widehat{Z}), \tag{5.6}$$

for every random variable Ω distributed according to an arbitrary probability distribution P_Ω and each distribution $q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}$. We begin by showing that

$$\mathbb{E}_\Omega\left[D_{\mathrm{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}} \| q_{\widehat{Z}|\mathbf{u_0}}\right)\right] \geqslant I_{\mathbf{u_0}}(\Omega;\widehat{Z}),$$

for any arbitrary distributions P_Ω and $q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}$. To this end, we use the following identities:

$$\begin{aligned}
\mathbb{E}_\Omega\left[D_{\mathrm{KL}}\left(q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}} \| q_{\widehat{Z}|\mathbf{u_0}}\right)\right] &= \mathbb{E}_\Omega \mathbb{E}_{q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}}\left(\log \frac{q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}}{q_{\widehat{Z}|\mathbf{u_0}}}\right) \\
&= \mathbb{E}_\Omega \mathbb{E}_{q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}}\left(\log \frac{q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}}{P_{\widehat{Z}}}\right) + D_{\mathrm{KL}}\left(P_{\widehat{Z}} \| q_{\widehat{Z}|\mathbf{u_0}}\right) \\
&= I_{\mathbf{u_0}}(\Omega;\widehat{Z}) + D_{\mathrm{KL}}\left(P_{\widehat{Z}} \| q_{\widehat{Z}|\mathbf{u_0}}\right) \geqslant I_{\mathbf{u_0}}(\Omega;\widehat{Z}),
\end{aligned} \tag{5.7}$$

where $P_{\widehat{Z}}$ denotes the marginal distribution of $q^{(\Omega)}_{\widehat{Z}|\mathbf{u_0}}$ w.r.t. P_Ω and the last inequality follows since the KL divergence is positive. Finally, it is easy to check that by selecting $q_{\widehat{Z}|\mathbf{u_0}} = P_{\widehat{Z}}$ the lower bound in (5.7) is achieved which proves the identity in expression (5.6). By taking the maximum overall probability distributions P_Ω at both sides of expression (5.6) the claim follows. □

From theory to our practical detector. According to our derivation in Eq. (5.5), the optimal detector turns out to be given by a mixture of the

$|\mathcal{K}|$ detectors belonging to the class $\mathcal{Q}$, with weights carefully optimized to maximize the mutual information between Ω and the predicted variable $\widehat{Z}$ for each detector in the class $\mathcal{Q}$. Using this key ingredient, it is straightforward to devise our optimal detector.

Definition 3. *For any* $0 \leqslant \gamma \leqslant 1$ *and a given* $\mathbf{x_0} \in \mathcal{X}$*, let us define the following detector* $\mathrm{D} : \mathbb{R}^d \to \{0, 1\}$*:*

$$\mathrm{D}(\mathbf{x}_0) \stackrel{\text{def}}{=} \mathbb{1}\left[q^{\star}_{\widehat{Z}|\mathbf{u_0}}(\hat{z} = 1|g^l_\theta(\mathbf{x_0})) > \gamma\right], \tag{5.8}$$

where $\mathbb{1}[\cdot]$ *is the indicator function.*

5.2 Experimental Results

We test our proposed solution by deploying it against the multi-armed adversarial attacks framework introduced in Chapter 4, and by evaluating its detection performance[1].

In our empirical evaluation, we assume that a third party provides us with four simple supervised detectors. Each of them is trained to detect a single specific kind of attack. This is a reasonable assumption, as many methods in the literature can successfully detect at least one type of attack and fail at detecting others. In addition, to emphasize the role played by the proposed method, these detectors are merely shallow networks (3 fully-connected layers with 256 nodes each), which are only allowed to observe the logits of the target classifier to distinguish between natural and adversarial samples. Due to their specifics, these individual shallow detectors are bound to perform very poorly, i.e. much worse than SOTA detectors, against attacks they have not been trained on, as shown in Fig. 5.3. This aspect enhances the value of our solution, which attains favorable performance by aggregating detectors that individually exhibit subpar performance w.r.t. SOTA adversarial examples detection methods.

5.2.1 Evaluation framework

Evaluation setup: MEAD. We consider all the attack algorithms mentioned in MEAD Chapter 4, and we group them by the corresponding norm and the

[1]The source code will be released upon acceptance of the paper this chapter is based on.

perturbation magnitude. For each natural sample and each gradient-based attack algorithm (i.e., FGSM, PGD or BIM), we create four adversarial examples, each corresponding to one of the loss functions described in Section 4.1.

Table 5.1 reports all the attacks in the multi-armed setting. Each cell corresponds to a group of attacks crafted according to the algorithm (reported in the cell), the associated norm (indicated by the column label) and perturbation magnitude (indicated by the row label) and one of the considered four loss functions. Thus, for example, when we consider L_∞ norm and $\varepsilon = 0.125$, the detector is evaluated on $4 + 4 + 4 + 1 = 13$ simultaneous adversarial attacks. Note that we discard the perturbed examples that do not fool the classifier as, by definition, they are neither natural nor adversarial.

Table 5.1: MEAD. Each cell corresponds to attacks simultaneously executed on the targeted classifier. Attacks created using all the losses in Section 4.1 are marked with *. Attacks such as SA and DF are not dependent on the choice for the loss but are equally considered as part of the multi-armed framework. Empty cells correspond to combinations of perturbation magnitude and norm constraint that are usually not considered in the literature.

	L_1	L_2	L_∞	No norm
$\varepsilon = 0.01$	-	CW2	-	-
$\varepsilon = 0.03125$	-	-	PGDi*,FGSM*,BIM*	-
$\varepsilon = 0.0625$	-	-	PGDi*,FGSM*,BIM*	-
$\varepsilon = 0.1$	-	HOP	-	-
$\varepsilon = 0.125$	-	PGD2*	PGDi*,FGSM*,BIM*,SA	-
$\varepsilon = 0.25$	-	PGD2*	PGDi*,FGSM*,BIM*	-
$\varepsilon = 0.3125$	-	PGD2*	PGDi*,FGSM*,BIM*,CWi	-
$\varepsilon = 0.5$	-	PGD2*	PGDi*,FGSM*,BIM*	-
$\varepsilon = 1$	-	PGD2*	-	-
$\varepsilon = 1.5$	-	PGD2*	-	-
$\varepsilon = 2$	-	PGD2*	-	-
$\varepsilon = 5$	PGD1*	-	-	-
$\varepsilon = 10$	PGD1*	-	-	-
$\varepsilon = 15$	PGD1*	-	-	-
$\varepsilon = 20$	PGD1*	-	-	-
$\varepsilon = 25$	PGD1*	-	-	-
$\varepsilon = 30$	PGD1*	-	-	-
$\varepsilon = 40$	PGD1*	-	-	-
No ε	-	DF	-	-
max. rotation = 30 max. translation = 8	-	-	-	STA

Evaluation metrics. Following the evaluation setup described above, for each sample and for each group of attacks corresponding to each cell in Table 5.1 we consider a detection

successful, i.e. a true positive, if and only if all the adversarial attacks are detected. Otherwise, we report a false negative. We use the classical definitions of *true negative* and *false positive* for the natural samples detection. This means that a true negative is a natural sample detected as natural, and a false positive is a natural sample detected as adversarial. We measure the performance of the detectors in terms of *i*) AUROC↑% [DG06b] (the *Area Under the Receiver Operating Characteristic curve*) which represents the ability of the detector to discriminate between adversarial and natural examples (higher is better); *ii*) FPR at 95 % TPR (FPR$\downarrow_{95\%}$%), i.e., the percentage of natural examples detected as adversarial when 95 % of the adversarial examples are detected (lower is better).

Datasets and pre-trained classifiers. We run our experiments on CIFAR10 [Kri09] and SVHN [NWC+11] image datasets. For both, the pre-trained target classifier is a ResNet-18 models that has been trained for 100 epochs, using SGD optimizer with a learning rate equal to 0.1, weight decay equal to 10^{-5}, and momentum equal to 0.9. The accuracy achieved by the classifiers on the original clean data is 99% for CIFAR10 and 100% for SVHN over the train split; 93.3% for CIFAR10 and 95.5% for SVHN over the test split.

Detectors. The proposed method aggregates four simple pre-trained detectors. The detectors are four fully-connected neural networks composed of 3 layers of 256 nodes each. All the detectors are trained for 100 epochs, using SGD optimizer with a learning rate of 0.01 and weight decay 0.0005. They are trained to distinguish between natural and adversarial examples created according to the PGD algorithm, under L_∞ norm constraint and perturbation magnitude $\varepsilon = 0.125$ for CIFAR10 and $\varepsilon = 0.25$ for SVHN. Each detector is trained on natural and adversarial examples generated using one of the loss functions mentioned in Section 4.1 (i.e., ACE Eq. (4.2), KL Eq. (4.3), FR Eq. (4.4), or Gini Eq. (4.5)) to craft its adversarial training samples. We want to point out that the purpose of this paper is not creating a new supervised detector but rather to show a method to aggregate a set of pre-trained detectors. Moreover, it is important to notice that either supervised and unsupervised methods can be added to or pool of experts (cf. Appendix C.2.1), provided that they output a

confidence on the input sample being or not an adversarial example. We further expand on the selection of the ε parameter of the adversarial examples used at training time in Appendix C.2.1 (cf. Tables C.3 and C.5).

NSS [KFHD20]. We compare the proposed method with NSS, which is the best among the supervised SOTA methods against multi-armed adversarial attacks (cf. Section 4.4). NSS characterizes the adversarial perturbations through the use of *natural scene statistics*, i.e., statistical properties that can be altered by the presence of adversarial perturbations. NSS is trained by using PGD algorithm, L_∞ norm constraint and perturbation magnitude $\varepsilon = 0.03125$ for CIFAR10 and $\varepsilon = 0.0625$ for SVHN. We further expand on the selection of the ε parameter of the adversarial examples used at training time in Tables C.2 and C.4 and Appendix C.2.1.

On the optimization of Eq. (5.5). For the optimization of Eq. (5.5), we rely on the `SciPy` [VGO+20] library, the `optimize` package, and the `minimize` function which uses the *Sequential Least Squares Programming* (SLSQP) algorithm to find the optimum. Further details can be found in Appendix C.1.

5.2.2 Discussion

We present the main experimental results to show the effectiveness of the proposed method for adversarial attack detection. Further discussion on these results, as well as additional experiments, can be found in Appendix C.2.

The *shallow* detectors

Figs. 5.1 to 5.3 provides a graphical interpretation of

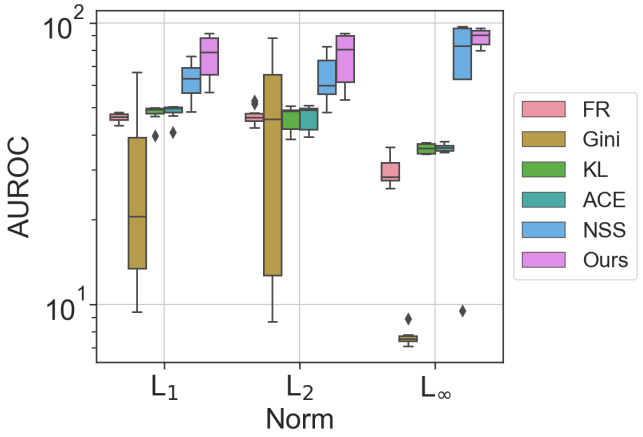


Figure 5.3: The *shallow* detectors are named after the loss function used to craft the attacks they are trained to detect. The SOTA method NSS outperforms all the individual shallow detectors. By aggregating shallow models, we can achieve a detector with comparable or better performance than SOTA.

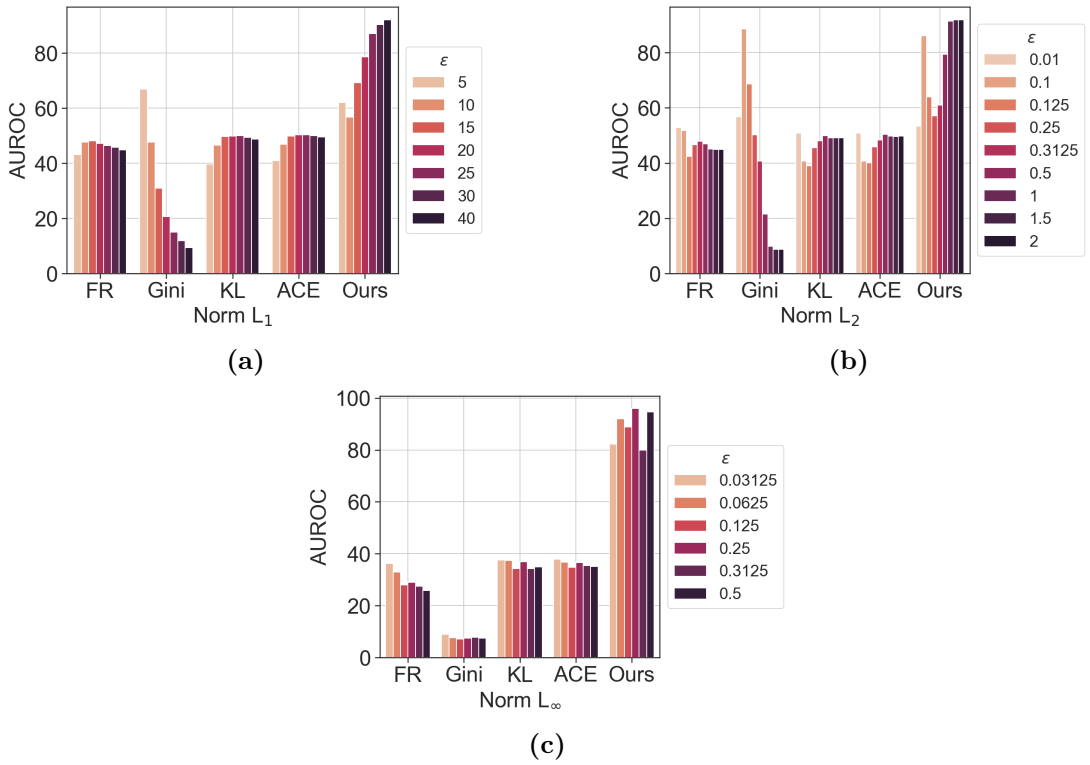
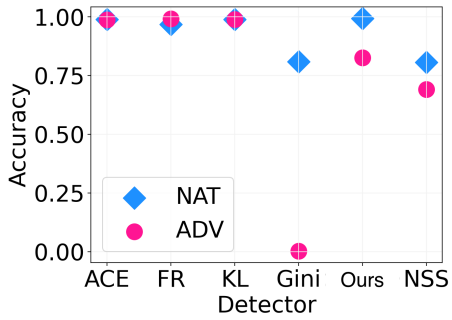
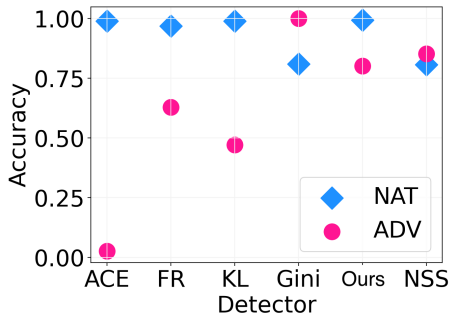


Figure 5.1: Performance of the various detectors grouped by L_p-norm and perturbation magnitude ε on CIFAR10. Each *shallow* detector is named after the loss function used to craft the attacks they it is trained to detect. The plot shows how our method consistently attains better performance than the single one on all the different adversarial attacks, supporting the claim of optimality in Section 5.1.

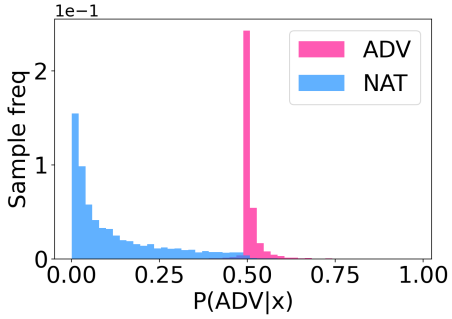
the detection performance when ResNet18, trained on CIFAR10, is the target classifier. The single detectors are named after the loss function used to craft the adversarial examples on which each detector is trained along with the natural samples. The main takeaway from Fig. 5.3 is the observation that, when considered individually, the shallow detectors are clearly subpar w.r.t. the state-of-the-art adversarial attacks detection mechanism. On the contrary, the aggregation provided by our method results in detection performance comparable to SOTA performance and, in some cases, outperforms well-established detection mechanisms. Figure 5.1 sheds light on the fact that the mixture of experts attained by our proposed method can consistently improve the detection of adversarial examples over several multi-armed attacks mounted using different norms and perturbation magnitudes.



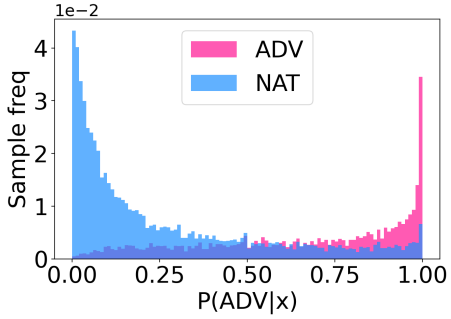
(a) Attacks crafted with PGD algorithm, the FR loss, $\varepsilon = 40$, and norm constraint L_1



(b) Attacks crafted with FGSM algorithm, the FR loss, $\varepsilon = 40$, and norm constraint L_∞



(c) Ours against attacks crafted with PGD algorithm, the FR loss, $\varepsilon = 40$, and norm constraint L_1



(d) NSS against attacks crafted with PGD algorithm, the FR loss, $\varepsilon = 40$, and norm constraint L_1

Figure 5.2: Discrimination performances. In Fig. 5.2a and Fig. 5.2b, the accuracies of the detectors on natural and adversarial examples; in Fig. 5.2c and Fig. 5.2d we show how the proposed method and NSS split the data samples. We report the results for detecting adversarial examples in pink and the results for detecting natural examples in blue.

One main takeaway of this paper is that, if we are provided with generally non-robust detectors whose performance is good only against a limited amount of attacks (as it is confirmed by Figs. 5.1 and 5.3), we can successfully aggregate them through the proposed method to obtain a consistently better detection.

In Fig. 5.2 we consider attacks crafted according to the PGD algorithm, the FR loss, $\varepsilon = 40$, and norm constraint L_1 (cf. Figs. 5.2a, 5.2c and 5.2d), and attacks crafted according to the FGSM algorithm, FR loss, $\varepsilon = 0.5$, and L_∞

norm in Fig. 5.2b. We also report the performance of the considered detectors in terms of detection accuracy over the natural examples in blue and the adversarial examples in pink. As we can observe, the individual detectors, which are named after the loss functions ACE, FR, KL, and Gini, exhibit different behaviors for the specific attack. In Fig. 5.2a, the Gini detector drastically fails at detecting the attack as its accuracy plummets to 0% on the adversarial examples. In the same way, the FR and KL detectors but mostly the ACE detector, perform poorly against FGSM (cf. Fig. 5.2b). On the contrary, our method, benefiting from the aggregation, obtains favorable results in both cases, confirming what we had previously observed.

The histograms in Figs. 5.2c and 5.2d show how the method we propose and NSS separate natural (blue) and adversarial examples (pink), respectively. The values along the horizontal axis represent the probability of being classified as adversarial, and the vertical axis represents the frequency of the samples within the bins. The detection error is proportional to the area of overlap between the blue and the pink histograms. Fig. 5.2c and Fig. 5.2d suggest that the proposed method achieves lower detection error on the considered attack, as it is confirmed in Table 5.2 where our proposed method attains 92.1 AUROC↑%, while NSS only achieves 76.1 AUROC↑% and. Additional plots are provided in Appendix C.2.4.

In particular, the performance attained by the proposed method is consistent across the larges part of the considered multi-armed adversarial attacks, as confirmed in Table 5.2 and Fig. 5.3.

Evaluation of the proposed aggregator in MEAD

On CIFAR10, our aggregator achieves maximum AUROC improvement w.r.t. NSS is 79.5 percentage points and happens for attacks under L_∞-norm constraint, $\varepsilon = 0.125$ and PGD*, FGSM*, BIM*, SA, i.e. when as many as 13 different simultaneous adversarial attacks are mounted. Similarly, for our proposed method the maximum attained FPR at 95% TPR improvement w.r.t. NSS is 90.3 percentage points and happens for attacks under L_∞-norm constraint, $\varepsilon = 0.5$ and PGD*, FGSM*, BIM*, i.e., when as many as 12 different simultaneous adversarial attacks are mounted. Our aggregator outperforms NSS in the case of the attacks with L_1

Table 5.2: Comparison between the proposed method and NSS on CIFAR10 and SVHN. The * symbol means the perturbation mechanism is executed in parallel four times starting from the same original clean sample, each time using one of the objective losses between ACE Eq. (4.2), KL Eq. (4.3), FR Eq. (4.4), Gini Eq. (4.5).

	CIFAR10				SVHN			
	NSS		Ours		NSS		Ours	
	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%
Norm L_1								
PGD1*								
$\varepsilon = 5$	48.5	94.2	**62.1**	**87.1**	40.2	91.3	**76.9**	**79.0**
$\varepsilon = 10$	54.0	**90.3**	**56.8**	90.6	36.9	91.3	**73.0**	**82.5**
$\varepsilon = 15$	58.8	86.8	**69.3**	**84.4**	35.6	91.3	**78.9**	**72.5**
$\varepsilon = 20$	63.5	82.3	**78.7**	**73.1**	36.1	91.3	**83.6**	**60.7**
$\varepsilon = 25$	67.7	77.2	**87.1**	**50.8**	37.8	91.3	**87.0**	**48.6**
$\varepsilon = 30$	71.4	73.4	**90.3**	**35.4**	39.8	91.3	**89.3**	**37.2**
$\varepsilon = 40$	76.1	67.3	**92.1**	**26.4**	43.1	91.3	**92.6**	**20.0**
Norm L_2								
PGD2*								
$\varepsilon = 0.125$	48.3	94.3	**63.9**	**85.4**	40.8	91.3	**80.2**	**74.5**
$\varepsilon = 0.25$	53.2	91.2	**57.1**	**90.5**	37.2	91.3	**74.0**	**81.7**
$\varepsilon = 0.3125$	55.8	89.2	**61.0**	**88.9**	36.1	91.3	**75.2**	**79.4**
$\varepsilon = 0.5$	63.3	82.6	**79.4**	**73.2**	35.9	91.3	**82.5**	**64.4**
$\varepsilon = 1$	76.4	67.5	**91.4**	**26.4**	42.5	91.3	**92.3**	**24.7**
$\varepsilon = 1.5$	81.0	63.0	**91.9**	**24.2**	46.3	91.3	**94.1**	**7.5**
$\varepsilon = 2$	82.6	62.3	**91.9**	**24.1**	49.8	91.3	**94.9**	**5.3**
DeepFool								
No ε	57.0	91.7	**81.9**	**54.8**	41.3	91.3	**94.9**	**12.0**
CW2								
$\varepsilon = 0.01$	**56.4**	**90.8**	53.4	92.2	41.0	**91.3**	**54.2**	92.0
HOP								
$\varepsilon = 0.1$	66.1	87.0	**86.1**	**49.1**	67.6	84.2	**96.0**	**10.2**
Norm L_∞								
PGDi*, FGSM*, BIM*								
$\varepsilon = 0.03125$	**83.0**	**55.3**	82.3	59.7	**86.3**	**46.9**	81.4	64.9
$\varepsilon = 0.0625$	**96.0**	**17.2**	92.0	29.6	88.9	**0.7**	**89.1**	33.3
$\varepsilon = 0.25$	**97.3**	**0.6**	95.9	8.8	51.6	88.9	**92.3**	**16.4**
$\varepsilon = 0.5$	82.5	100.0	**94.6**	**9.7**	46.7	86.7	**92.9**	**14.4**
PGDi*, FGSM*, BIM*, SA								
$\varepsilon = 0.125$	9.4	99.9	**88.9**	**40.8**	32.9	91.3	**89.2**	**29.1**
PGDi*, FGSM*, BIM*, CWi								
$\varepsilon = 0.3125$	63.2	99.1	**80.0**	**61.1**	41.3	91.3	**88.2**	**33.1**
No norm								
STA								
No ε	**88.5**	**38.8**	82.7	52.4	**91.2**	**0.2**	90.2	23.2

Table 5.3: Comparison between Ours and Ours+NSS on CIFAR10. The * symbol means the perturbation mechanism is executed in parallel four times starting from the same original clean sample, each time using one of the objective losses between ACE Eq. (4.2), KL Eq. (4.3), FR Eq. (4.4), Gini Eq. (4.5). We focus only in the cases in which the proposed method is outperformed from the corresponding competitors.

	CIFAR10			
	Ours		Ours+NSS	
	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%
Norm L_2				
CW2				
$\varepsilon = 0.01$	53.4	92.2	**54.1**	**91.3**
Norm L_∞				
PGDi*, FGSM*, BIM*				
$\varepsilon = 0.03125$	82.3	59.7	**89.9**	**34.4**
$\varepsilon = 0.0625$	92.0	29.6	**96.4**	**9.0**
$\varepsilon = 0.25$	95.9	8.8	**96.7**	**3.5**
No norm				
STA				
No ε	82.7	52.4	**87.3**	**35.4**

and L_2 norm, regardless of the algorithm or the perturbation magnitude, and in the case of L_∞ norm with large perturbations. However, for the attacks with L_∞ norm and small ε, although the proposed method's performance is comparable to that of NSS, we notice a slight degradation. To shed light on this, we remind that individual detectors aggregated are based on the classifier's logits; NSS, on the other hand, extracts natural scene statistics from the inputs. This more sophisticated technique makes NSS perform well when tested on attacks with similar ε and the same norm as the ones seen at training time. Similar conclusions can be drawn for the results on SVHN (cf. Table 5.2).

Table 5.3 shows the modularity of the proposed method when SOTA detection methods, NSS (a) and FS (b), are plugged in as a fifth detector. We test Ours+NSS on the attacks on which our aggregator was outperformed by the competitors. In all the cases, Ours+NSS outperforms "Ours" either in terms of AUROC and FPR. In most of the cases, Ours+NSS is also better than the individual competitor. In Appendix C.2.1 we provide further insights on this by showing that the same behavior is observed when we plug a SOTA unsupervised

method as fifth detector in our pool.

Evaluation of the proposed aggregator in the non-simultaneous setting

In these experiments, we move from the simultaneous adversarial attack scenario to one where the different detectors are aggregated to detect one single attack at a time, as usually done in the literature. We report the complete results Table 5.4. Crucially, these experiments show that ensemble detectors can also improve the performance for specific attacks. In particular, we would like to draw attention to the fact that we outperform NSS in the vast majority of the cases. Moreover, we achieve a maximum gain of 82.8 percentage points in terms of AUROC↑% (cf. SA attack) and 97.6 percentage points in terms of $\text{FPR}\downarrow_{95\%}\%$ (cf. FGSM with $\varepsilon = 0.5$ attack). On the other side, the competitor outperforms our proposed method only in a few cases, achieving a maximum gain of 5.9 percentage points in terms of AUROC↑% and 27.4 percentage points in terms of $\text{FPR}\downarrow_{95\%}\%$ (cf. FGSM with ε=0.03125 attack in both the cases), and these gains are much lower than those obtained by the proposed method.

5.3 Final remarks

We introduced a new method to tackle the multi-armed adversarial attacks introduced in Mead Chapter 4. We formalized the multi-armed attack detection problem as a minimax cross-entropy risk and derived a surrogate loss function. Based on this, we characterized our optimal soft-detector, which results in a mixture of experts, as the solution to a minimax problem. Our empirical results show that aggregating simple detectors using our method consistently improves detection performance. The achieved performance is comparable and, in a large set of cases, better than the best state-of-the-art (SOTA) method in the multi-armed attack scenarios. Our method has two key benefits: it is modular, allowing existing and future methods to be integrated, and it is general, able to recognize adversarial examples from various attack algorithms and loss functions. Additionally, our aggregator can potentially be extended to aggregate both supervised and unsupervised SOTA adversarial detection methods.

Limitations of the proposed method come from the fact it relies on a collection of detectors whose expertise is combined to obtain a more robust adversarial

Table 5.4: The proposed method and NSS in the non-simultaneous setting. The column names ACE, KL, FR, and Gini denote the loss function used to craft the attacks. HOP, DeepFool, CW2, and STA attacks have already been considered individually in Table 5.2.

	CIFAR10			
	Ours AUROC↑% (FPR↓$_{95\%}$%) – NSS AUROC↑% (FPR↓$_{95\%}$%)			
	ACE	KL	FR	Gini
PGD1				
$\varepsilon = 5$	**66.2 (83.6)** – 49.9 (93.5)	**64.2 (85.7)** – 49.6 (93.0)	**63.0 (87.1)** – 49.9 (93.3)	**80.7 (58.4)** – 50.3 (93.2)
$\varepsilon = 10$	**62.6 (87.5)** – 56.9 (88.4)	**62.3 (88.2)** – 56.6 (88.3)	**63.1 (86.5)** – 57.0 (88.1)	**86.9 (46.0)** – 57.1 (88.8)
$\varepsilon = 15$	**74.2 (81.4)** – 63.1 (83.0)	**75.2 (80.6)** – 62.8 (83.1)	**75.3 (79.4)** – 63.2 (82.5)	**90.0 (31.1)** – 63.5 (84.0)
$\varepsilon = 20$	**86.8 (65.3)** – 68.5 (77.1)	**87.5 (63.1)** – 68.1 (77.3)	**86.9 (63.3)** – 68.7 (76.4)	**91.7 (31.2)** – 69.9 (77.6)
$\varepsilon = 25$	**93.9 (38.4)** – 73.1 (71.1)	**94.3 (36.2)** – 72.7 (71.8)	**93.7 (41.1)** – 73.4 (70.9)	**92.3 (28.9)** – 75.0 (71.4)
$\varepsilon = 30$	**97.1 (12.3)** – 77.1 (64.5)	**97.2 (12.6)** – 76.8 (65.1)	**96.8 (15.9)** – 77.4 (65.2)	**92.6 (27.9)** – 78.6 (67.3)
$\varepsilon = 40$	**98.9 (1.0)** – 83.5 (52.7)	**99.0 (1.0)** – 83.3 (53.5)	**98.8 (1.0)** – 83.6 (52.7)	**92.7 (27.4)** – 80.1 (64.9)
PGD2				
$\varepsilon = .125$	**67.9 (81.1)** – 49.5 (93.8)	**65.4 (84.3)** – 49.1 (93.5)	**63.9 (86.6)** – 49.6 (93.5)	**80.6 (58.4)** – 49.5 (94.3)
$\varepsilon = .25$	**62.3 (87.5)** – 55.9 (89.1)	**62.1 (88.0)** – 55.6 (89.2)	**62.6 (87.6)** – 55.8 (89.4)	**86.7 (46.5)** – 55.9 (89.8)
$\varepsilon = .3125$	**66.5 (86.1)** – 59.4 (86.5)	**67.0 (85.9)** – 59.0 (86.6)	**67.8 (84.8)** – 59.3 (86.6)	**88.4 (42.2)** – 59.3 (87.7)
$\varepsilon = .5$	**86.4 (67.1)** – 68.3 (77.4)	**87.2 (64.5)** – 68.0 (77.4)	**86.7 (64.0)** – 68.4 (77.2)	**91.4 (31.4)** – 69.0 (78.7)
$\varepsilon = 1$	**98.9 (0.9)** – 84.4 (50.6)	**98.9 (0.9)** – 84.3 (50.5)	**98.8 (0.9)** – 84.7 (50.7)	**92.5 (27.2)** – 79.3 (66.8)
$\varepsilon = 1.5$	**99.2 (0.9)** – 92.8 (28.7)	**99.3 (0.9)** – 92.7 (28.9)	**99.3 (0.7)** – 93.0 (27.3)	**92.5 (27.2)** – 79.5 (66.5)
$\varepsilon = 2$	**99.3 (0.8)** – 96.8 (13.9)	**99.3 (0.8)** – 96.9 (13.1)	**99.3 (0.9)** – 95.9 (17.2)	**92.5 (27.2)** – 79.5 (66.5)
PGDi				
$\varepsilon = .03125$	**99.1 (0.9)** – 92.3 (31.0)	**99.1 (0.9)** – 92.1 (31.9)	**99.0 (0.9)** – 92.2 (30.7)	**94.8 (21.5)** – 89.0 (44.0)
$\varepsilon = .0625$	**99.3 (0.8)** – 99.1 (3.3)	**99.3 (0.8)** – 99.1 (3.3)	**99.3 (0.8)** – 99.1 (3.6)	97.4 **(8.0)** – **98.1** (8.1)
$\varepsilon = .125$	99.3 (0.7) – **99.7 (0.6)**	99.3 (0.9) – **99.7 (0.6)**	99.3 (0.8) – **99.6 (0.6)**	97.3 (7.3) – **99.6 (0.6)**
$\varepsilon = .25$	99.3 (0.7) – **99.7 (0.6)**	99.3 (0.9) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	97.1 (7.3) – **99.6 (0.6)**
$\varepsilon = .3125$	99.3 (0.9) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	97.1 (7.4) – **99.7 (0.6)**
$\varepsilon = .5$	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	97.1 (7.3) – **99.6 (0.6)**
FGSM				
$\varepsilon = .03125$	89.2 (47.5) – **94.1 (26.7)**	91.3 (40.6) – **94.0 (27.0)**	92.6 (34.1) – **96.8 (15.0)**	90.7 (42.7) – **96.6 (15.3)**
$\varepsilon = .0625$	96.4 (18.5) – **99.4 (1.3)**	96.2 (18.7) – **99.4 (1.4)**	97.6 (10.3) – **99.6 (0.6)**	97.4 (11.9) – **99.6 (0.6)**
$\varepsilon = .125$	99.3 (3.4) – **99.7 (0.6)**	99.1 (4.3) – **99.7 (0.6)**	99.3 (2.5) – **99.5 (0.6)**	99.3 (2.4) – **99.5 (0.6)**
$\varepsilon = .25$	**99.8 (0.6)** – 99.7 **(0.6)**	**99.7** (0.8) – **99.7 (0.6)**	**99.6** (1.1) – 97.9 **(0.6)**	**99.6** (1.1) – 97.7 **(0.6)**
$\varepsilon = .3125$	**99.7** (0.9) – **99.7 (0.6)**	**99.7** (0.9) – **99.7 (0.6)**	**99.5** (1.5) – 95.8 **(0.6)**	**99.5** (1.5) – 95.6 **(0.6)**
$\varepsilon = .5$	99.0 (4.9) – **99.7 (0.6)**	99.2 (2.7) – **99.7 (0.6)**	**99.2 (2.4)** – 84.9 (100.0)	**99.2 (2.4)** – 84.8 (100.0)
BIM				
$\varepsilon = .03125$	**98.3 (4.6)** – 90.3 (37.7)	**98.3 (4.4)** – 90.2 (38.1)	**97.8 (7.2)** – 90.5 (37.0)	**92.2 (32.6)** – 88.2 (45.1)
$\varepsilon = .0625$	**99.4 (0.8)** – 98.2 (7.5)	**99.4 (0.9)** – 98.2 (7.5)	**99.4 (0.8)** – 98.3 (7.3)	96.6 (13.1) – **97.3 (12.9)**
$\varepsilon = .125$	99.3 (0.9) – **99.6 (0.7)**	99.3 (0.9) – **99.7 (0.7)**	99.3 (0.8) – **99.6 (0.7)**	97.8 (6.9) – **99.3 (1.9)**
$\varepsilon = .25$	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.9) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	97.4 (7.2) – **99.6 (0.6)**
$\varepsilon = .3125$	99.3 (0.9) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.9) – **99.7 (0.6)**	97.1 (7.4) – **99.7 (0.6)**
$\varepsilon = .5$	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	99.3 (0.8) – **99.7 (0.6)**	96.3 (7.3) – **99.7 (0.6)**
SA				
$\varepsilon = .125$	**91.2 (39.6)** – 9.4 (99.9)	**91.2 (39.6)** – 9.4 (99.9)	**91.2 (39.6)** – 9.4 (99.9)	**91.2 (39.6)** – 9.4 (99.9)
CWi				
$\varepsilon = .3125$	**80.7 (60.8)** – 64.6 (89.8)	**80.7 (60.8)** – 64.6 (89.8)	**80.7 (60.8)** – 64.6 (89.8)	**80.7 (60.8)** – 64.6 (89.8)

detection. Such models could be potentially poisoned by a malicious actor, drastically reducing the aggregator's reliability. We think this could have a potentially severe societal impact if the proposed method happened to be deployed with no additional checks on the quality of the available detectors.

CHAPTER 6

Conclusion of the Thesis

In this thesis, we have investigated how to formulate security problems as binary hypothesis testing.

We first addressed the problem of *misclassification detection*. Given an input sample and a pre-trained classifier, we want to understand whether the input sample comes from the distribution of the correctly classified samples or the incorrectly classified ones. Based on the soft-probabilities associated with the example, the detector approximates the probability of the classifier outputting the wrong class. The prediction will be rejected if the computed score exceeds a certain threshold. Through simulations on image and text datasets, we have demonstrated the superiority of the proposed detector over the SOTA techniques. Furthermore, we distinguished two scenarios: the Totally Black-Box (TBB) scenario, in which the detector has access only to the final soft-probabilities, and the Partially Black-Box (PBB) scenario, in which the detector has access to the logits and can perform input pre-processing.

We have then moved on to the problem of *multi-armed adversarial attack detection*. In this case, the goal of the detector is to check whether the input sample is natural or has been adversarially perturbed according to *some* strategy. We refer to this setting as 'multi-armed' as in the classical ('single-armed') detection setting the methods are generally validated by assuming a single attack strategy at a time. We formalize the problem as a minimax cross-entropy risk. Based on this formulation, we derive a surrogate loss function and use it to characterize our optimal soft-detector leading to our aggregator of detectors' decisions. Experimentally, we have shown: (i) the classical framework led to an overopti-

mistic assessment of the detectors' performance; (ii) the proposed aggregator of detectors' decisions represents a valid (and the first) solution to the multi-armed setting.

The line of research we have pursued in this work offers many opportunities for future work. For instance, as to the misclassification detection topic, we are thoroughly studying how our framework adapts to the case of image segmentation. In this context, the detector is asked to detect errors in the predicted mask outputs from the pre-trained model. The analysis can be conducted either by pixel or region. Interestingly, the results so far indicate that the proposed detector can recognize when a region should be segmented but is not present in the predicted mask. In particular, we are dealing with a critical safety system like the one of medical diagnosis by performing simulations on datasets such as Automated Cardiac Diagnosis Challenge [BLZ+18], and Brain Tumor Segmentation Challenge [MJB+15], to cite a few. Moreover, a straightforward extension of the proposed method would be for regression tasks. Currently, the detector we presented relies on the soft probability outputs from the classifier. Therefore, it can not be used as it is also with regression models. A possibility in this sense would be to quantize the output space and treat each of the bins into which the space is divided as a possible class. Anyway, one issue in this sense would be how to assign to each of the bins a probability.

A more ambitious goal would be to apply our detector aggregator to topics beyond simultaneous adversarial attack detection. As long as the detector outputs can be interpreted as a probability distribution across two categories, any existing or future supervised or unsupervised method can be combined using our proposed approach, making the aggregator a new ensemble technique. An example of this extension is intrusion detection, where an improved detection framework is highly desired, particularly with the use of ensemble learners [TL21].

Bibliography

[ACFH20] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIII*, volume 12368 of *Lecture Notes in Computer Science*, pages 484–501. Springer, 2020.

[ACW18] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[AF21] Sahar Abdelnabi and Mario Fritz. What's in the box: Deflecting adversarial attacks by randomly deploying adversarially-disjoint models. In Trent Jaeger and Zhiyun Qian, editors, *MTD@CCS 2021: Proceedings of the 8th ACM Workshop on Moving Target Defense, Virtual Event, Republic of Korea, 15 November 2021*, pages 3–12. ACM, 2021.

[AHFD22] Ahmed Aldahdooh, Wassim Hamidouche, Sid Ahmed Fezza, and Olivier Deforges. Adversarial example detection for dnn models: A review and experimental comparison. *Artificial Intelligence Review*, 2022.

[Asm03] *Steady-State Properties of GI/G/1*, pages 266–301. Springer New York, New York, NY, 2003.

[BGMS17] Michael Backes, Manuel Gomez-Rodriguez, Praveen Manoharan, and Bartlomiej Surma. Reconciling privacy and utility in continuous-time diffusion networks. In *30th IEEE Computer Security Foundations Symposium, CSF 2017, Santa Barbara, CA, USA, August 21-25, 2017*, pages 292–304. IEEE Computer Society, 2017.

[BHP+21] Oliver Bryniarski, Nabeel Hingun, Pedro Pachuca, Vincent Wang, and Nicholas Carlini. Evading adversarial example detection defenses with orthogonal projected gradient descent. *CoRR*, abs/2106.15023, 2021.

[BLZ+18] Olivier Bernard, Alain Lalande, Clément Zotti, Frederic Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Ángel González Ballester, Gerard Sanroma, Sandy Napel, Steffen E. Petersen, Georgios Tziritas, Elias Grinias, Mahendra Khened, Alex Varghese, Ganapathy Krishnamurthi, Marc-Michel Rohé, Xavier Pennec, Maxime Sermesant, Fabian Isensee, Paul Jaeger, Klaus H. Maier-Hein, Peter M. Full, Ivo Wolf, Sandy Engelhardt, Christian F. Baumgartner, Lisa M. Koch, Jelmer M. Wolterink, Ivana Isgum, Yeonggul Jang, Yoonmi Hong, Jay Patravali, Shubham Jain, Olivier Humbert, and Pierre-Marc Jodoin. Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: Is the problem solved? *IEEE Trans. Medical Imaging*, 37(11):2514–2525, 2018.

[BRY98] Andrew R. Barron, Jorma Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *IEEE Trans. Inf. Theory*, 44(6):2743–2760, 1998.

[CBB19] Francesco Crecchi, Davide Bacciu, and Battista Biggio. Detecting black-box adversarial examples through nonlinear dimensionality reduction. In *27th European Symposium on Artificial Neural Networks, ESANN 2019, Bruges, Belgium, April 24-26, 2019*, 2019.

[CH20] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In

Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, pages 2206–2216. PMLR, 2020.

[CJW20] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*, pages 1277–1294. IEEE, 2020.

[CW17a] Nicholas Carlini and David A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In Bhavani Thuraisingham, Battista Biggio, David Mandell Freeman, Brad Miller, and Arunesh Sinha, editors, *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 3–14. ACM, 2017.

[CW17b] Nicholas Carlini and David A. Wagner. Magnet and "efficient defenses against adversarial attacks" are not robust to adversarial examples. *CoRR*, abs/1711.08478, 2017.

[CW17c] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society, 2017.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[DG06a] Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh,*

Pennsylvania, USA, June 25-29, 2006, volume 148, pages 233–240, 2006.

[DG06b] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[DT18] Terrance DeVries and Graham W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *CoRR*, abs/1802.04865, 2018.

[ETT+19] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1802–1811. PMLR, 2019.

[FCSG17] Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. Detecting adversarial samples from artifacts. *CoRR*, abs/1703.00410, 2017.

[GE17] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4878–4887, 2017.

[GE19] Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97, pages 2151–2159, 2019.

[GGP21] Federica Granese, Daniele Gorla, and Catuscia Palamidessi. Enhanced models for privacy and utility in continuous-time diffusion networks. *Int. J. Inf. Sec.*, 20(5):763–782, 2021.

[GJG+22] Daniele Gorla, Louis Jalouzot, Federica Granese, Catuscia Palamidessi, and Pablo Piantanida. On the (im)possibility of estimating various notions of differential privacy. *CoRR*, abs/2208.14414, 2022.

[GKS21] Aditya Gangrade, Anil Kag, and Venkatesh Saligrama. Selective classification via one-sided prediction. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 2179–2187. PMLR, 2021.

[GPR+22] Federica Granese, Marine Picot, Marco Romanelli, Francisco Messina, and Pablo Piantanida. MEAD: A multi-armed approach for evaluation of adversarial examples detectors. *CoRR*, abs/2206.15415, 2022.

[GPSW17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70, pages 1321–1330, 2017.

[GRG+21] Federica Granese, Marco Romanelli, Daniele Gorla, Catuscia Palamidessi, and Pablo Piantanida. DOCTOR: A simple method for detecting misclassification errors. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 5669–5681, 2021.

[GRGP23] Federica Granese, Marco Romanelli, Siddharth Garg, and Pablo Piantanida. A minimax approach against multi-armed adversarial attacks detection, 2023.

[GSS15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[HAB19] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 41–50. Computer Vision Foundation / IEEE, 2019.

[HCSS21] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ML safety. *CoRR*, abs/2109.13916, 2021.

[HLW16] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[HMD19] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

[HPS16] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3315–3323, 2016.

[HSJK20] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized ODIN: detecting out-of-distribution image without learning

from out-of-distribution data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10948–10957, 2020.

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[IST+19] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 125–136, 2019.

[JWX17] Qixiang Zhang Jiayu Wu and Guoxi Xu. Tiny imagenet challenge. Technical report, 2017.

[KE16] Volodymyr Kuleshov and Stefano Ermon. Reliable confidence estimation via online learning. *CoRR*, abs/1607.03594, 2016.

[KFHD20] Anouar Kherchouche, Sid Ahmed Fezza, Wassim Hamidouche, and Olivier Déforges. Detection of adversarial examples in deep neural networks with natural scene statistics. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pages 1–7. IEEE, 2020.

[KGB] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*.

[KHH20] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5436–5446. PMLR, 2020.

[KL15] Volodymyr Kuleshov and Percy Liang. Calibrated structured prediction. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3474–3482, 2015.

[Kri09] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[LCB10] Yann LeCun, Corinna Cortes, and Chris Burges. Mnist handwritten digit database, 2010.

[LLLS18a] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7167–7177, 2018.

[LLLS18b] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7167–7177, 2018.

[LLS18] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[LWOL20] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[MC17] Dongyu Meng and Hao Chen. Magnet: A two-pronged defense against adversarial examples. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 135–147. ACM, 2017.

[MDP+11] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, June 2011.

[MFF16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2574–2582. IEEE Computer Society, 2016.

[MJB+15] Bjoern H. Menze, András Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin S. Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, Levente Lanczi, Elizabeth R. Gerstner, Marc-André Weber, Tal Arbel, Brian B.

Avants, Nicholas Ayache, Patricia Buendia, D. Louis Collins, Nicolas Cordier, Jason J. Corso, Antonio Criminisi, Tilak Das, Herve Delingette, Çagatay Demiralp, Christopher R. Durst, Michel Dojat, Senan Doyle, Joana Festa, Florence Forbes, Ezequiel Geremia, Ben Glocker, Polina Golland, Xiaotao Guo, Andac Hamamci, Khan M. Iftekharuddin, Raj Jena, Nigel M. John, Ender Konukoglu, Danial Lashkari, José Antonio Mariz, Raphael Meier, Sérgio Pereira, Doina Precup, Stephen J. Price, Tammy Riklin Raviv, Syed M. S. Reza, Michael T. Ryan, Duygu Sarikaya, Lawrence H. Schwartz, Hoo-Chang Shin, Jamie Shotton, Carlos A. Silva, Nuno J. Sousa, Nagesh K. Subbanna, Gábor Székely, Thomas J. Taylor, Owen M. Thomas, Nicholas J. Tustison, Gözde B. Ünal, Flor Vasseur, Max Wintermark, Dong Hye Ye, Liang Zhao, Binsheng Zhao, Darko Zikic, Marcel Prastawa, Mauricio Reyes, and Koen Van Leemput. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans. Medical Imaging*, 34(10):1993–2024, 2015.

[MLW+18] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[MMS+18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[NLM19] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Confer-*

ence on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 188–197, 2019.

[NWC+11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[Pla00] John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.*, 10, 06 2000.

[PMB+22] M. Picot, F. Messina, M. Boudiaf, F. Labeau, I. Ben Ayed, and P. Piantanida. Adversarial robustness via fisher-rao regularization. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2022.

[PSCvdH22] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, 54(2):38:1–38:38, 2022.

[Rud94] Daniel L. Ruderman. The statistics of natural images. *Network: Computation In Neural Systems*, 5:517–548, 1994.

[SHS19] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6976–6987. Computer Vision Foundation / IEEE, 2019.

[SLSZ03] A. Srivastava, A. B. Lee, Eero P. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *J. Math. Imaging Vis.*, 18(1):17–33, 2003.

[SMB21] Adi Shamir, Odelia Melamed, and Oriel BenShmuel. The dimpled manifold model of adversarial examples in machine learning. *CoRR*, abs/2106.10151, 2021.

[SST+18] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5019–5031, 2018.

[SZS+14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[TCBM20] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[TG16] Thomas Tanay and Lewis D. Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *CoRR*, abs/1608.07690, 2016.

[TKP+18] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[TL21] Bayu Adhi Tama and Sung Hoon Lim. Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation. *Comput. Sci. Rev.*, 39:100357, 2021.

[Tsy08] Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[Vap95] Vladimir Naumovich Vapni. *The Nature of Statistical Learning Theory*. Springer, 1995.

[vEH14] T. van Erven and P. Harremos. Rényi Divergence and Kullback-Leibler Divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[VGO+20] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[VI20] Vladimir Vapnik and Rauf Izmailov. Complete statistical theory of learning: learning using statistical invariants. In *Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications*, volume 128, pages 4–40, 2020.

[vN28] John von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928.

[WDS+20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empir-*

ical Methods in Natural Language Processing: System Demonstrations, pages 38–45, oct 2020.

[XEQ18] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society, 2018.

[XTG+20] Cihang Xie, Mingxing Tan, Boqing Gong, Alan L. Yuille, and Quoc V. Le. Smooth adversarial training. *CoRR*, abs/2006.14536, 2020.

[XYF+22] Jiancong Xiao, Liusha Yang, Yanbo Fan, Jue Wang, and Zhi-Quan Luo. Understanding adversarial robustness against on-manifold adversarial examples. *CoRR*, abs/2210.00430, 2022.

[YBTV21] Chengyuan Yao, Pavol Bielik, Petar Tsankov, and Martin Vechev. Automated discovery of adaptive attacks on adversarial defenses. *Advances in Neural Information Processing Systems*, 34:26858–26870, 2021.

[ZYJ+19] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 1–11, 2019.

APPENDIX A

Appendix to Chapter 3

A.1 Proofs

The following section shows the proofs for Proposition (1), Proposition (2) and Inequalities (2.3).

A.1.1 Proof of Proposition 1

We recall the definition of the total variation distance when applied to distributions P, Q on a set $\mathcal{X} \subseteq \mathbb{R}^d$ and the Scheffé's identity, Lemma 2.1 in [Tsy08]:

$$\|P - Q\|_{\text{TV}} \stackrel{\text{def}}{=} \sup_{\mathcal{A} \in \mathcal{B}^d} |P(\mathcal{A}) - Q(\mathcal{A})| = \frac{1}{2} \int |p_X(\mathbf{x}) - q_X(\mathbf{x})| d\mu(\mathbf{x}) \tag{A.1}$$

with respect to a base measure μ, where $\mathcal{B}^d$ denotes the class of all Borel sets on $\mathbb{R}^d$.

First of all, we prove the equality for $\gamma = 1$. Let us denote with $\mathcal{A}^\star \equiv \mathcal{A}(1)$ and $\mathcal{A}^{\star c} \equiv \mathcal{A}^c(1)$ the optimal decision regions from (3.6). Let $\epsilon_0(\mathcal{A}^\star)$ and $\epsilon_1(\mathcal{A}^{\star c})$

be the Type-I and Type-II errors, respectively. Then,

$$\begin{aligned}
\epsilon_0(\mathcal{A}^\star) + \epsilon_1(\mathcal{A}^{\star c}) &= \int_{\mathcal{A}^\star} p_{X|E}(\mathbf{x}|0)d\mathbf{x} + \int_{\mathcal{A}^{\star c}} p_{X|E}(\mathbf{x}|1)d\mathbf{x} \\
&= \int_{\mathcal{A}^\star} \min\Big\{p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1)\Big\}d\mathbf{x} \\
&+ \int_{\mathcal{A}^{\star c}} \min\Big\{p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1)\Big\}d\mathbf{x} \\
&= \int_{\mathcal{X}} \min\Big\{p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1)\Big\}d\mathbf{x} \\
&= 1 - \big\|p_{X|E=1} - p_{X|E=0}\big\|_{\mathrm{TV}}\,, \qquad \text{(A.2)}
\end{aligned}$$

where the last identity follows by applying Scheffé's identity (A.1). From the last identity in (A.2) and any decision region $\mathcal{A} \subseteq \mathcal{X}$, we have

$$\begin{aligned}
1 - \big\|p_{X|E=1} - p_{X|E=0}\big\|_{\mathrm{TV}} &= \int_{\mathcal{X}} \min\Big\{p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1)\Big\}d\mathbf{x} \\
&= \int_{\mathcal{A}} \min\Big\{p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1)\Big\}d\mathbf{x} \\
&+ \int_{\mathcal{A}^c} \min\Big\{p_{X|E}(\mathbf{x}|0)\,,\, p_{X|E}(\mathbf{x}|1)\Big\}d\mathbf{x} \\
&\leqslant \int_{\mathcal{A}} p_{X|E}(\mathbf{x}|0)d\mathbf{x} + \int_{\mathcal{A}^c} p_{X|E}(\mathbf{x}|1)d\mathbf{x} \\
&= \epsilon_0(\mathcal{A}) + \epsilon_1(\mathcal{A}^c). \qquad \text{(A.3)}
\end{aligned}$$

It remains to show the last statement related to the Bayesian error of the test. Assume that $p_E(1) = p_E(0) = 1/2$. By using the last identity in (A.2), we have

$$\begin{aligned}
\frac{1}{2}\Big[1 - \big\|p_{X|E=1} - p_{X|E=0}\big\|_{\mathrm{TV}}\Big] &= \frac{1}{2}\int_{\mathcal{X}} \min\Big\{p_{X|E}(\mathbf{x}|0), p_{X|E}(\mathbf{x}|1)\Big\}d\mathbf{x} \\
&= \int_{\mathcal{X}} \min\Big\{p_{XE}(\mathbf{x}, E=0), p_{XE}(\mathbf{x}, E=1)\Big\}d\mathbf{x} \\
&= \mathbb{E}_X\left[\min\Big\{p_{E|X}(0|\mathbf{X}), p_{E|X}(1|\mathbf{X})\Big\}\right] \\
&= \frac{1}{2}\big[\epsilon_0(\mathcal{A}^\star) + \epsilon_1(\mathcal{A}^{\star c})\big] \\
&\equiv \inf_D \Pr\{D(\mathbf{X}) \neq E\}\,, \qquad \text{(A.4)}
\end{aligned}$$

where the last identity follow by the definition of the decision regions in (3.6).

A.1.2 Proof of Proposition 2

We begin by showing that

$$\begin{aligned}
|\widehat{\mathrm{Pe}}(\mathbf{x}) - \mathrm{Pe}(\mathbf{x})| &= \Big|\mathbb{E}\big[\mathbb{1}[\widehat{Y} \neq g_\theta(\mathbf{x})]\big|\mathbf{x}\big] - \mathbb{E}\big[\mathbb{1}[Y \neq g_\theta(\mathbf{x})]\big|\mathbf{x}\big]\Big| \\
&= \left|\sum_{\{y\in\mathcal{Y}\,|\,y\neq g_\theta(\mathbf{x})\}} \big[p_{\widehat{Y}|X}(y|\mathbf{x};\theta) - p_{Y|X}(y|\mathbf{x})\big]\right| \\
&\leqslant \sum_{\{y\in\mathcal{Y}\,|\,y\neq g_\theta(\mathbf{x})\}} \Big|p_{\widehat{Y}|X}(y|\mathbf{x};\theta) - p_{Y|X}(y|\mathbf{x})\Big| \\
&\leqslant \sum_{y\in\mathcal{Y}} \Big|p_{\widehat{Y}|X}(y|\mathbf{x};\theta) - p_{Y|X}(y|\mathbf{x})\Big| \\
&\leqslant 2\left\|p_{\widehat{Y}|X}(\cdot|\mathbf{x};\theta) - p_{Y|X}(\cdot|\mathbf{x})\right\|_{\mathrm{TV}} \\
&\leqslant 2\sqrt{2\mathrm{KL}\left(p_{Y|\mathbf{x}}\|p_{\widehat{Y}|\mathbf{x}}\right)}, \qquad \text{(A.5)}
\end{aligned}$$

where $\|\cdot\|_{\mathrm{TV}}$ denotes the *Total Variation distance*, $\mathrm{KL}(\cdot\|\cdot)$ is the *Kullback–Leibler divergence* and the last step is due to *Pinsker's inequality*. On the other hand,

$$\begin{aligned}
1 - \widehat{\mathrm{g}}(\mathbf{x}) &= 1 - \sum_{y\in\mathcal{Y}} p^2_{\widehat{Y}|X}(y|\mathbf{x};\theta) \\
&= 1 - \mathbb{E}_{\widehat{Y}|X}\left[p_{\widehat{Y}|X}(\widehat{Y}|\mathbf{x};\theta)|\mathbf{x}\right] \\
&\geqslant 1 - \mathbb{E}_{\widehat{Y}|X}\left[\max_{y\in\mathcal{Y}} p_{\widehat{Y}|X}(y|\mathbf{x};\theta)\big|\mathbf{x}\right] \\
&= 1 - \max_{y\in\mathcal{Y}} p_{\widehat{Y}|X}(y|\mathbf{x};\theta) \\
&\equiv \widehat{\mathrm{Pe}}(\mathbf{x}). \qquad \text{(A.6)}
\end{aligned}$$

Similarly,

$$\begin{aligned}
\widehat{\mathrm{g}}(\mathbf{x}) = \sum_{y\in\mathcal{Y}} p^2_{\widehat{Y}|X}(y|\mathbf{x};\theta) &= p^2_{\widehat{Y}|X}(y^\star|\mathbf{x};\theta) + \sum_{y\neq y^\star} p^2_{\widehat{Y}|X}(y|\mathbf{x};\theta) \\
&\geqslant \max_{y\in\mathcal{Y}} p^2_{\widehat{Y}|X}(y|\mathbf{x};\theta) \\
&\equiv \left(1 - \widehat{\mathrm{Pe}}(\mathbf{x})\right)^2, \qquad \text{(A.7)}
\end{aligned}$$

where $y^\star = \arg\max_{y\in\mathcal{Y}} p_{\widehat{Y}|X}(y|\mathbf{x};\theta)$. By replacing expressions (A.6) and (A.7) in (A.5) we obtained the desired inequalities, which concludes the proof.

A.1.3 Proof of Inequalities in (2.3)

The event can be decomposed as follows:

$$\big\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\big\} \equiv \big\{Y \neq \widehat{Y}\big\} \cap \Big\{\big\{\widehat{Y} = g_\theta(\mathbf{x})\big\} \text{ or } \big\{Y = g_\theta(\mathbf{x})\big\}|\mathbf{x}\Big\} \tag{A.8}$$

for all $\mathbf{x} \in \mathcal{X}$. Thus,

$$\big\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\big\} \subseteq \big\{Y \neq \widehat{Y}|\mathbf{x}\big\}, \tag{A.9}$$

$$\big\{Y \neq \widehat{Y}\big\} \cap \big\{Y \neq g_\theta(\mathbf{x})\big|\mathbf{x}\big\} \subseteq \big\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\big\}, \tag{A.10}$$

$$\big\{Y \neq \widehat{Y}\big\} \cap \big\{\widehat{Y} \neq g_\theta(\mathbf{x})|\mathbf{x}\big\} \subseteq \big\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\big\}, \tag{A.11}$$

which imply

$$\Pr\big(\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})|\mathbf{x}\}\big) \leqslant \Pr\big(\{\widehat{Y} \neq Y\}|\mathbf{x}\big), \tag{A.12}$$

$$\mathrm{Pe}(\mathbf{x}) - \Pr\big(\{\widehat{Y} = Y\}|\mathbf{x}\big) \leqslant \Pr\big(\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})\}\big|\mathbf{x}\big), \tag{A.13}$$

$$\widehat{\mathrm{Pe}}(\mathbf{x}) - \Pr\big(\{\widehat{Y} = Y\}|\mathbf{x}\big) \leqslant \Pr\big(\{\widehat{E}(\mathbf{x}) \neq E(\mathbf{x})\}\big|\mathbf{x}\big), \tag{A.14}$$

for all $\mathbf{x} \in \mathcal{X}$, where the last inequalities follows by noticing that $\Pr(\mathcal{A} \cap \mathcal{B}) \geqslant \Pr(\mathcal{A}) - \Pr(\mathcal{B}^c)$ for arbitrary measurable sets $\mathcal{A}, \mathcal{B} \subset \mathcal{X}$. This concludes the proof of these inequalities.

A.2 Logistic Regression and Gaussian Model

Throughout this section we test DOCTOR in a controlled setting were all the involved distributions are known. We refer to that setting as *logistic regression and Gaussian model* since we collect data points from Gaussians distributions and we test on the logistic regression setup.

A.2.1 Theoretical analysis

Let $\mathcal{X} = \mathbb{R}^d$ be the feature space and $\mathcal{Y} = \{-1, 1\}$ be the label space. We focus on a binary classification task in which $\mathbf{X} \sim \mathcal{N}(y\mu, \sigma^2 I)$ and $Y \sim \mathcal{U}(\mathcal{Y})$, where

$\mu \in \mathbb{R}^n$ is the mean vector, $\sigma^2 > 0$ is the variance and I is the identity matrix and $\mathcal{U}(\mathcal{Y})$ denotes the uniform distribution over $\mathcal{Y}$. For a fixed $\theta \in \mathbb{R}^d$, consider $f_\theta : \mathcal{X} \to \mathcal{Y}$ s.t. $f_\theta(\mathbf{x}) = \text{sign}(\text{sigmoid}(\mathbf{x}^T\theta) - 1/2)$. For a given $\mathbf{x} \in \mathcal{X}$, we adapt to the current setting the definition of $E(\mathbf{x})$ in Chapter 3 as follows:

$$\mathbb{1}\left[Y \neq f_\theta(\mathbf{x})\right] = \mathbb{1}\left[Y \cdot \text{sign}\left(\text{sigmoid}\left(\mathbf{x}^T\theta\right) - \frac{1}{2}\right) < 0\right]. \tag{A.15}$$

Let us denote by $\mathbb{1}\left[y \neq f_\theta(\mathbf{x})\right]$ the realization of the random variable $E(\mathbf{x})$. We can compute the probability of classification error $\text{Pe}(\mathbf{x})$ in (2.1) w.r.t. the true class posterior probabilities:

$$\begin{aligned}
\text{Pe}(\mathbf{x}) = \mathbb{E}\Big[\mathbb{1}\left[Y \neq f_\theta(\mathbf{x})\right]|\mathbf{x}\Big] &= \sum_{y\in\mathcal{Y}} \mathbb{1}\left[y \neq f_\theta(\mathbf{x})\right] \cdot \frac{p_{\mathbf{X}|Y}(\mathbf{x}|y)P_Y(y)}{p_{\mathbf{X}}(\mathbf{x})} \\
&= \sum_{y\in\mathcal{Y}} \mathbb{1}\left[y \neq f_\theta(\mathbf{x})\right] \cdot \frac{\frac{1}{2}\mathcal{N}(\mathbf{x}; y\mu, \sigma^2 I)}{\frac{1}{2}\sum_{y'\in\mathcal{Y}}\mathcal{N}(\mathbf{x}; y'\mu, \sigma^2 I)} \\
&= \frac{\sum_{y\in\mathcal{Y}} \mathbb{1}\left[y \neq f_\theta(\mathbf{x})\right] \cdot \mathcal{N}(\mathbf{x}; y\mu, \sigma^2 I)}{\sum_{y\in\mathcal{Y}}\mathcal{N}(\mathbf{x}; y\mu, \sigma^2 I)}.
\end{aligned} \tag{A.16}$$

Following (3.7), the decision region corresponding to the most powerful discriminator for the logistic regression and the Gaussian model are given by

$$\mathcal{A}(\gamma) = \left\{\mathbf{x} \in \mathcal{X} : \frac{\sum_{y\in\mathcal{Y}} \mathbb{1}\left[y \neq f_\theta(\mathbf{x})\right] \cdot \mathcal{N}(\mathbf{x}, y\mu, \sigma^2 I)}{\sum_{y\in\mathcal{Y}} \mathbb{1}\left[y = f_\theta(\mathbf{x})\right] \cdot \mathcal{N}(\mathbf{x}, y\mu, \sigma^2 I)} > \gamma\right\}. \tag{A.17}$$

We are now able to state the optimal discriminator for this setting.

Definition 4 (Optimal discriminator for the logistic regression and the Gaussian model)**.** *For any $0 < \gamma < \infty$ and $\mathbf{x} \in \mathcal{X}$, the optimal discriminator follows as:*

$$D^\star(\mathbf{x}, \gamma) \stackrel{\text{def}}{=} \mathbb{1}\left[\sum_{y\in\mathcal{Y}} \mathbb{1}\left[y \neq f_\theta(\mathbf{x})\right] \cdot \mathcal{N}(\mathbf{x}, y\mu, \sigma^2 I) > \gamma \cdot \sum_{y\in\mathcal{Y}} \mathbb{1}\left[y = f_\theta(\mathbf{x})\right] \cdot \mathcal{N}(\mathbf{x}, y\mu, \sigma^2 I)\right]. \tag{A.18}$$

Since we cannot analytically evaluate Proposition 1, we proceed numerically in the next experiment.

A.2.2 Experiments

In this section, we will numerically evaluate Proposition 1 via empirical estimates of Type-I and Type-II errors in expressions Eq. (3.2). Note that unlike Section 3.4, in this case, all the involved distributions are known, and hence it is also possible to compute the *true posterior distribution* $p_{Y|X}$.

We adopt the same notation as in Section 3.1.2 for DOCTOR, i.e., D_α, and D_β according to according to expressions Eq. (3.11). $D^\star$, as in Definition 4, denotes the optimal discriminator.

Table A.1: Accuracy on the test set: f_{θ_i} for $i = 1, \ldots, 8$ represents the i-th model in $\mathcal{F}$, f_{avg} is the arithmetic mean of the accuracy over each $f_{\theta_i} \in \mathcal{F}$. The value $f^\star_{avg}$ represents the accuracy Bayesian classifier averaged on the test set corresponding to the 8 splits. We show results for both standard deviations, namely $\sigma = 2$ and $\sigma = 4$.

CLASSIFIER	ACCURACY%	
	$\sigma = 2$	$\sigma = 4$
f_{θ_1}	82	65
f_{θ_2}	83	77
f_{θ_3}	82	77
f_{θ_4}	82	76
f_{θ_5}	83	76
f_{θ_6}	81	66
f_{θ_7}	82	76
f_{θ_8}	83	83
f_{avg}	82	74
$f^\star_{avg}$	83	78

Experimental setup and evaluation metrics

Dataset. We create a synthetic dataset that consists of 5000 data points drawn from $\mathcal{N}_0 \overset{\text{def}}{=} \mathcal{N}(\mu_0, \sigma^2 I)$ and 5000 data points drawn from $\mathcal{N}_1 \overset{\text{def}}{=} \mathcal{N}(\mu_1, \sigma^2 I)$, where $\mu_0 = [-1 \ -1]$, $\mu_1 = [1\ 1]$. We consider two values for sigma, namely $\sigma = 2$ and $\sigma = 4$. These values produce two different distributions which will let us showcase the advantages of DOCTOR. To each data point $\mathbf{x}$ is assigned as class 0 or 1 depending on whether $\mathbf{x} \sim \mathcal{N}_0$ or $\mathbf{x} \sim \mathcal{N}_1$, respectively. The aforementioned dataset is divided into a training set, i.e. $\mathcal{D}_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ where $n = 6700$, and a testing set, i.e. $\mathcal{T}_m = \{(\mathbf{x}_{n+1}, y_{n+1}), \ldots, (\mathbf{x}_{n+m}, y_{n+m})\}$ where $m = 3300$.

Training configuration. We use a linear classifier, with one hidden layer, sigmoid activation function and binary cross entropy loss. The neural network is trained with gradient descent considering learning rate $r = 0.1$. Specifically, we train our network for 5 epochs. We randomly split our dataset 8 times, each time keeping n samples to train, and m to test. We consider the same model

architecture (described above) for each split and we come up with 8 different binary discriminators $\mathcal{F} = \{f_{\theta_1}, \ldots, f_{\theta_8}\}$.

Since in this example all the involved distributions are known, we compute the optimal predictor, i.e., the Bayes classifier, and we denote it with $f^\star$. The value $f^\star_{avg}$ reported in Table A.1, represents its accuracy averaged on the test set corresponding to the 8 splits.

Accuracy of trained networks. In Table A.1 the accuracy of $f^\star$ and the models in $\mathcal{F}$ on the test set.

Evaluation metric. We consider the same metric as in Section 3.4.2.

Numerical evaluation of Proposition 1

To evaluate Proposition 1 we proceed in a Monte Carlo fashion by computing Type-I and Type-II errors for each of the networks in $\mathcal{F}$ and then averaging over the results. Schematically, consider any $f_{\theta_i} \in \mathcal{F}$ and $\gamma = 1$, we compute:

1. $\mathcal{A}_i \stackrel{\text{def}}{=} \mathcal{A}_i(1)$ as defined in Eq. (A.17) and its complement $\mathcal{A}_i^c$.
2. For each classifier $f_{\theta_i} \in \mathcal{F}$, $\mathcal{T}_{E=1;\theta_i} \stackrel{\text{def}}{=} \{(\mathbf{x}, y) \in \mathcal{T}_m \mid y \neq f_{\theta_i}(\mathbf{x})\}$ represents the set of mis-classified test samples, and $\mathcal{T}_{E=0;\theta_i} \stackrel{\text{def}}{=} \{(\mathbf{x}, y) \in \mathcal{T}_m \mid y = f_{\theta_i}(\mathbf{x})\}$ is the set of correctly classified test samples.
3. $\mathcal{FR}_i \stackrel{\text{def}}{=} \{(\mathbf{x}, y) \in \mathcal{T}_{E=0;\theta_i} : \mathbf{x} \in \mathcal{A}_i\}$, $\mathcal{TR}_i \stackrel{\text{def}}{=} \{(\mathbf{x}, y) \in \mathcal{T}_{E=1;\theta_i} : \mathbf{x} \in \mathcal{A}_i\}$, $\mathcal{FA}_i \stackrel{\text{def}}{=} \{(\mathbf{x}, y) \in \mathcal{T}_{E=1;\theta_i} : \mathbf{x} \in \mathcal{A}_i^c\}$ and $\mathcal{TA}_i \stackrel{\text{def}}{=} \{(\mathbf{x}, y) \in \mathcal{T}_{E=0;\theta_i} : \mathbf{x} \in \mathcal{A}_i^c\}$, i.e. the set of false rejections, true rejections, false acceptances and true acceptance, respectively.
4. $\epsilon_0(\mathcal{A}_i) \stackrel{\text{def}}{=} \frac{|\mathcal{FR}_i|}{|\mathcal{T}_{E=0;\theta_i}|}$ and $\epsilon_1(\mathcal{A}_i^c) \stackrel{\text{def}}{=} \frac{|\mathcal{FA}_i|}{|\mathcal{T}_{E=1;\theta_i}|}$, i.e. Type-I and Type-II errors.

At the end of $|\mathcal{F}|$ iterations, we empirically estimate Type-I and Type-II errors of Proposition 1 as follows

$$\epsilon_0(\mathcal{A}) \approx \frac{1}{|\mathcal{F}|}\sum_{i=1}^{|\mathcal{F}|}\epsilon_0(\mathcal{A}_i) = 0.0607 \quad \text{and} \quad \epsilon_1(\mathcal{A}^c) \approx \frac{1}{|\mathcal{F}|}\sum_{i=1}^{|\mathcal{F}|}\epsilon_1(\mathcal{A}_i^c) = 0.7389.$$

FRR versus TRR

We present the experimental results obtained by running experiments similar to those described in Section 3.4 considering the experimental setup in A.2.2 in TBB.

Table A.2: AUROCs: the values for D_α, D_β, SR, and ODIN correspond to the results for the thick lines in Fig. A.1. $D^\star$ and ODIN$^\star \equiv$ SR* are obtained using $p_{Y|X}$.

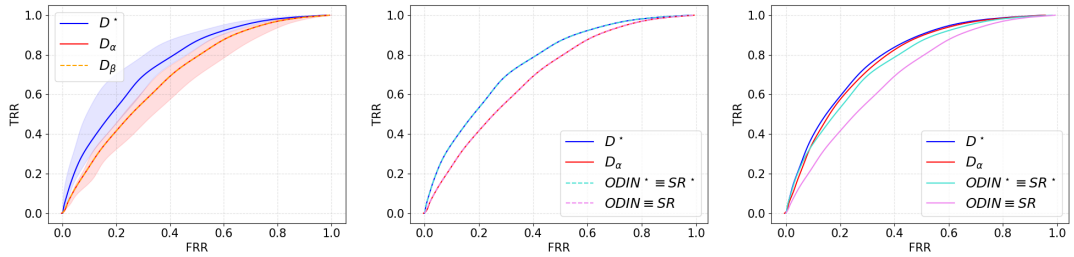
	AUROC %				
σ	$D^\star$	D_α	D_β	SR $\equiv$ ODIN	SR$^\star \equiv$ ODIN*
2	**76**	70	70	70	76
4	**79**	78	78	70	76

In addition to the usual discriminators, we will consider the optimal discriminator $D^\star$, as in Definition 4.

DOCTOR: comparison between $D^\star$, D_α and D_β. Let us present the result obtained with DOCTOR showing how $D^\star$, Eq. (A.18), works compared to D_α and D_β in Eq. (3.11) when they have to decide whether to trust or not the decision made by a classifier. We test the discriminators on the dataset constructed as in Appendix A.2.2 by considering $\sigma = 2$. Let us analyze Fig. A.1a: we apply each discriminator to all the classifiers in $\mathcal{F}$. The colored areas represent the obtained ROCs. Inside each area, the mean ROC is represented by the thick line. D_α and D_β reach the same results as the colored areas, and the thick lines are overlapped. For a given $\mathbf{x} \in \mathcal{X}$, we recall that $D^\star$ uses $\mathrm{Pe}(\mathbf{x})$ Eq. (2.1) whilst D_α and D_β uses $1 - \widehat{\mathrm{g}}(\mathbf{x})$ Eq. (3.8) and $\widehat{\mathrm{Pe}}(\mathbf{x})$ Eq. (2.2), respectively. $D^\star$ always outperforms both D_α and D_β since it relies on the probability of classification error based on $p_{Y|X}$ while D_α and D_β use $p_{\widehat{Y}|X}$.

Comparison between $D^\star$, D_α, ODIN and SR. We conclude this section by investigating how our competitors, namely ODIN and SR, work in this setting.

From now on, we will put ODIN $\equiv$ SR to mean that the two methods coincide (remember we set $T = 1$ and $\epsilon = 0$ for all the simulations). We show the results of the comparison in Fig. A.1: Fig. A.1b considers data points from $\mathcal{N}(y\mu, 2^2 I)$ whilst Fig. A.1c consider data points from $\mathcal{N}(y\mu, 4^2 I)$. If in Fig. A.1b we cannot see an advantage in using D_α in place of SR, the situation is totally different in Fig. A.1c, where $D^\star$ and D_α clearly outperform the competitors. We would like to recall that DOCTOR uses all the softmax output while SR only uses the maximum value of the softmax output.



(a) $\sigma = 2$ **(b)** $\sigma = 2$ **(c)** $\sigma = 4$

Figure A.1: ROC curves for $D^\star$, D_α and D_β, respectively. We denote by SR* the softmax response method based on $p_{Y|X}$. Since in this case $T = 1$ and $\epsilon = 0$, SR $\equiv$ ODIN as well as SR$^\star \equiv$ ODIN*. (a) We apply each discriminator to all the classifiers in $\mathcal{F}$. The obtained ROCs are represented by the colored areas. Inside each area the mean ROC is represented by the thick line. Orange and red areas completely overlap as well as the mean ROC. $D^\star$ always outperforms both D_α and D_β as expected. In (b) $D^\star$ and SR* overlap (as also D_α and SR), instead in (c) where $\sigma = 4$ SR discards useful information and indeed both $D^\star$ and D_α outperform SR.

A.3 Supplementary Results of Section 3.4

A.3.1 Experimental environment

We run each experiment on a machine equipped with an Intel(R) Xeon(R) CPU E5-2623 v4, 2.60GHz clock frequency, and a GeForce GTX 1080 Ti GPU. The execution time for the execution the tests are the following (interval size 10000):

TBB. D_α: 12.5 s. D_β: 13.6 s. SR: 15.9 s. MHLNB: 15.9 s.

PBB: D_α: 13 s. D_β: 25.7 s. ODIN: 14.7 s. MHLNB: 32.22 s.

A.3.2 On the input pre-processing in DOCTOR

In the following we further study DOCTOR-specific input pre-processing techniques allowed under PBB. We focus on D_β since for D_α the reasoning is the same. Formally, let $\mathbf{x}_0 \in \mathcal{X}$ be a testing sample. We are looking for the minimum way to perturb the input such that the discriminator value at $\mathbf{x}_0$ is increased:

$$r^* = \min_{r \text{ s.t. } \|r\|_\infty \leqslant \epsilon} -\log\left(\frac{\widehat{\text{Pe}}(\mathbf{x}_0 + r)}{1 - \widehat{\text{Pe}}(\mathbf{x}_0 + r)}\right),$$

or equivalently, we are looking to the sample $\widetilde{\mathbf{x}}_0^\beta$ in the ϵ-ball around $\mathbf{x}_0$ which maximize the discriminator value at $\widetilde{\mathbf{x}}_0^\beta$:

$$\widetilde{\mathbf{x}}_0^\beta = \mathbf{x}_0 - \epsilon \times \operatorname{sign}\left[-\nabla_{\mathbf{x}_0} \log\left(\frac{\widehat{\mathrm{Pe}}(\mathbf{x}_0)}{1-\widehat{\mathrm{Pe}}(\mathbf{x}_0)}\right)\right].$$

Note that, because of Eq. (2.1)

$$\begin{aligned}
-\log\left(\frac{\widehat{\mathrm{Pe}}(\mathbf{x}_0)}{1-\widehat{\mathrm{Pe}}(\mathbf{x}_0)}\right) &= -\log\left(\frac{1-p_{\hat{Y}|X}(g_\theta(\mathbf{x}_0)|\mathbf{x}_0;\theta)}{p_{\hat{Y}|X}(g_\theta(\mathbf{x}_0)|\mathbf{x}_0;\theta)}\right) \\
&= -\log(1-p_{\hat{Y}|X}(g_\theta(\mathbf{x}_0)|\mathbf{x}_0;\theta)) + \log(p_{\hat{Y}|X}(g_\theta(\mathbf{x}_0)|\mathbf{x}_0;\theta)) \\
&= -\log(1-p_{\hat{Y}|X}(g_\theta(\mathbf{x}_0)|\mathbf{x}_0;\theta)) - \log \mathrm{SODIN}(\mathbf{x}_0).
\end{aligned}$$

A.3.3 On the effect the intervals considered for γ, δ and ζ have on the AUROC computation

Let us consider the AUROC as a performance measure for the discriminators. The computation of the AUROC of D_α, as well as those of ODIN and SR, heavily depend on the choice of the range values for the decision region thresholds. In the following paragraph, we will discuss how we chose these ranges, namely $\gamma \in \Gamma_{D_\alpha \text{ or } D_\beta} \subseteq \mathbb{R}$, $\delta \in \Delta_{\mathrm{ODIN \text{ or } SR}} \subseteq [0,1]$ and $\zeta \in Z_{\mathrm{MHLNB}} \subseteq \mathbb{R}$. In the experiments of Section 3.4, we therefore proceed by fixing the aforementioned ranges as follows:

$$\Gamma_{D_\alpha} \overset{\text{def}}{=} \left[\min_{(\mathbf{x},y)\in\mathcal{T}_m} \frac{1-\widehat{\mathrm{g}}(\mathbf{x})}{\widehat{\mathrm{g}}(\mathbf{x})}, \max_{(\mathbf{x},y)\in\mathcal{T}_m} \frac{1-\widehat{\mathrm{g}}(\mathbf{x})}{\widehat{\mathrm{g}}(\mathbf{x})}\right], \tag{A.19}$$

$$\Gamma_{D_\beta} \overset{\text{def}}{=} \left[\min_{(\mathbf{x},y)\in\mathcal{T}_m} \frac{\widehat{\mathrm{Pe}}(\mathbf{x})}{1-\widehat{\mathrm{Pe}}(\mathbf{x})}, \max_{(\mathbf{x},y)\in\mathcal{T}_m} \frac{\widehat{\mathrm{Pe}}(\mathbf{x})}{1-\widehat{\mathrm{Pe}}(\mathbf{x})}\right], \tag{A.20}$$

$$\Delta_{\mathrm{ODIN}} \overset{\text{def}}{=} \left[\min_{(\mathbf{x},y)\in\mathcal{T}_m} \mathrm{SODIN}(\mathbf{x}), \max_{(\mathbf{x},y)\in\mathcal{T}_m} \mathrm{SODIN}(\mathbf{x})\right], \tag{A.21}$$

$$\Delta_{\mathrm{SR}} \overset{\text{def}}{=} \left[\min_{(\mathbf{x},y)\in\mathcal{T}_m} \mathrm{SR}(\mathbf{x}), \max_{(\mathbf{x},y)\in\mathcal{T}_m} \mathrm{SR}(\mathbf{x})\right], \tag{A.22}$$

$$Z_{\mathrm{MHLNB}} \overset{\text{def}}{=} \left[\min_{(\mathbf{x},y)\in\mathcal{T}_m} \mathrm{M}(\mathbf{x}), \max_{(\mathbf{x},y)\in\mathcal{T}_m} \mathrm{M}(\mathbf{x})\right]. \tag{A.23}$$

Secondly, we fix the number of values to consider in $\Gamma_{D_\alpha \text{ or } D_\beta}$, $\Delta_{\mathrm{ODIN \text{ or } SR}}$ and Z_{MHLNB}: we test the AUROCs for CIFAR10 for different values of the size of

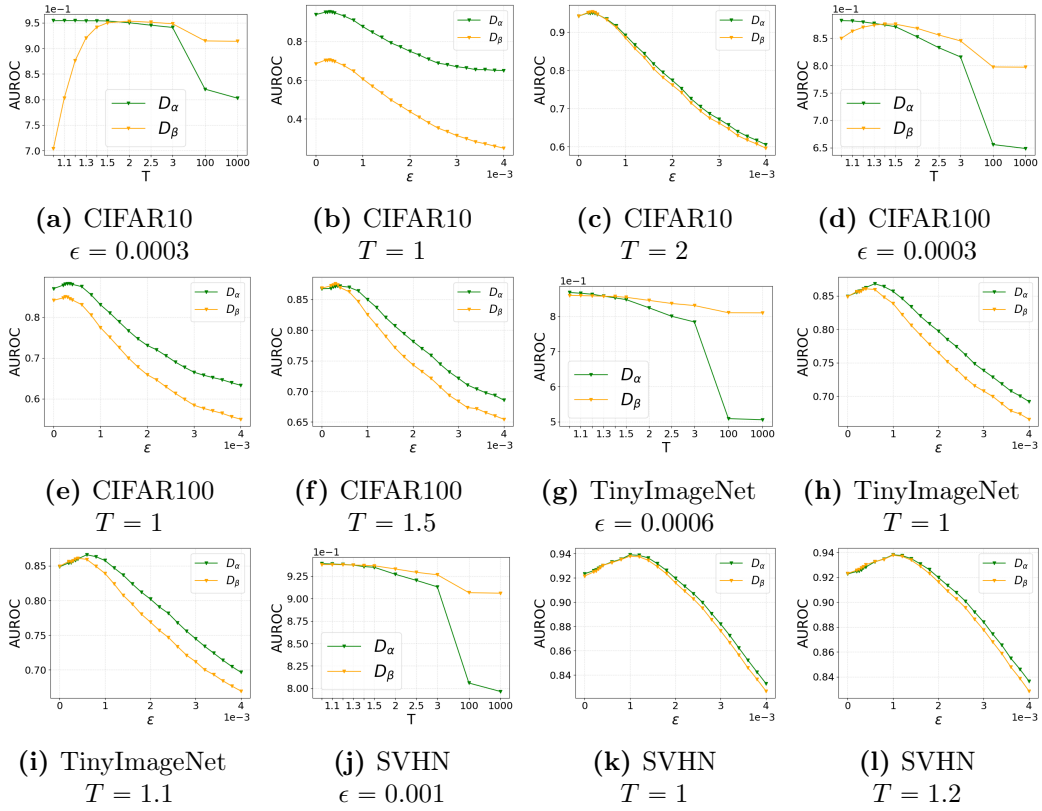
Table A.3: The effects of varying the number of thresholds.AUROCs and FRR at 95% TRR obtained via D_α, D_β, ODIN, SR and MHLNB for CIFAR10 considering different size for $\Gamma_{D_\alpha \text{ or } D_\beta}$, $\Delta_{\text{ODIN or SR}}$ and Z_{MHLNB} in both TBB and PBB. The column INTERVAL SIZE represents the number of equidistant values considered in the sets defined in (A.19), (A.20), (A.21), (A.22) and in (A.23), respectively.

INTERVAL SIZE	METHOD	TBB		PBB	
		AUROC	FRR (95 % TRR)	AUROC	FRR (95 % TRR)
10	D_α	69.8	91.6	77.4	88.4
	D_β	50	69.7	79.8	86.2
	ODIN	75.7	89.3	81.4	85.4
	SR	75.7	89.3	-	-
	MHLNB	76.6	88.8	83.2	47.1
100	D_α	85.1	80.6	92.5	42.6
	D_β	61.8	63.4	94.1	13.8
	ODIN	88	73.5	91.5	49.9
	SR	88	73.5	-	-
	MHLNB	88.3	72.6	84.4	44.6
1000	D_α	91.3	53.1	94.7	13.8
	D_β	66.5	48.3	94.8	13.4
	ODIN	92.5	28.9	94	18.3
	SR	92.5	28.9	-	-
	MHLNB	92.2	35.3	84.4	44.5
10000	D_α	93.7	18.4	95.2	13.9
	D_β	68.5	18.6	94.8	13.4
	ODIN	93.9	18	94.2	18.4
	SR	93.9	18	-	-
	MHLNB	92.1	31	84.4	44.6

$\Gamma_{D_\alpha \text{ or } D_\beta}$, $\Delta_{\text{ODIN or SR}}$ and Z_{MHLNB} in both TBB and PBB scenarios. The results are collected in Table A.3. Let us denote by I a generic interval between the ones of Eq. (A.19), Eq. (A.20), Eq. (A.21), Eq. (A.22) and Eq. (A.23), throughout the experiments we set the size of I to $(\max I - \min I) * 10000$.

A.3.4 Additional plots and results

In the next sections, we show graphically the set of results obtained from the experiments in Section 3.4.3. We first specify the range of values for the parameters T and ϵ considered throughout the experiments. For temperature scaling, T is selected among $\{1,\ 1.1,\ 1.2,\ 1.3,\ 1.4,\ 1.5,\ 2,\ 2.5,\ 3,\ 100,\ 1000\}$, whilst for input pre-processing, ϵ is selected among $\{0,\ .0002,\ .00025,\ .0003,\ .00035,\ .0004,\ .0006,\ .0008,\ .001,\ .0012,\ .0014,\ .0016,\ .0018,\ .002,\ .0022,\ .0024,\ .0026,\ .0028,\ .003,\ .0032,\ .0034,\ .0036,\ .0038,\ .004\}$.



(a) CIFAR10 $\epsilon = 0.0003$ **(b)** CIFAR10 $T = 1$ **(c)** CIFAR10 $T = 2$ **(d)** CIFAR100 $\epsilon = 0.0003$

(e) CIFAR100 $T = 1$ **(f)** CIFAR100 $T = 1.5$ **(g)** TinyImageNet $\epsilon = 0.0006$ **(h)** TinyImageNet $T = 1$

(i) TinyImageNet $T = 1.1$ **(j)** SVHN $\epsilon = 0.001$ **(k)** SVHN $T = 1$ **(l)** SVHN $T = 1.2$

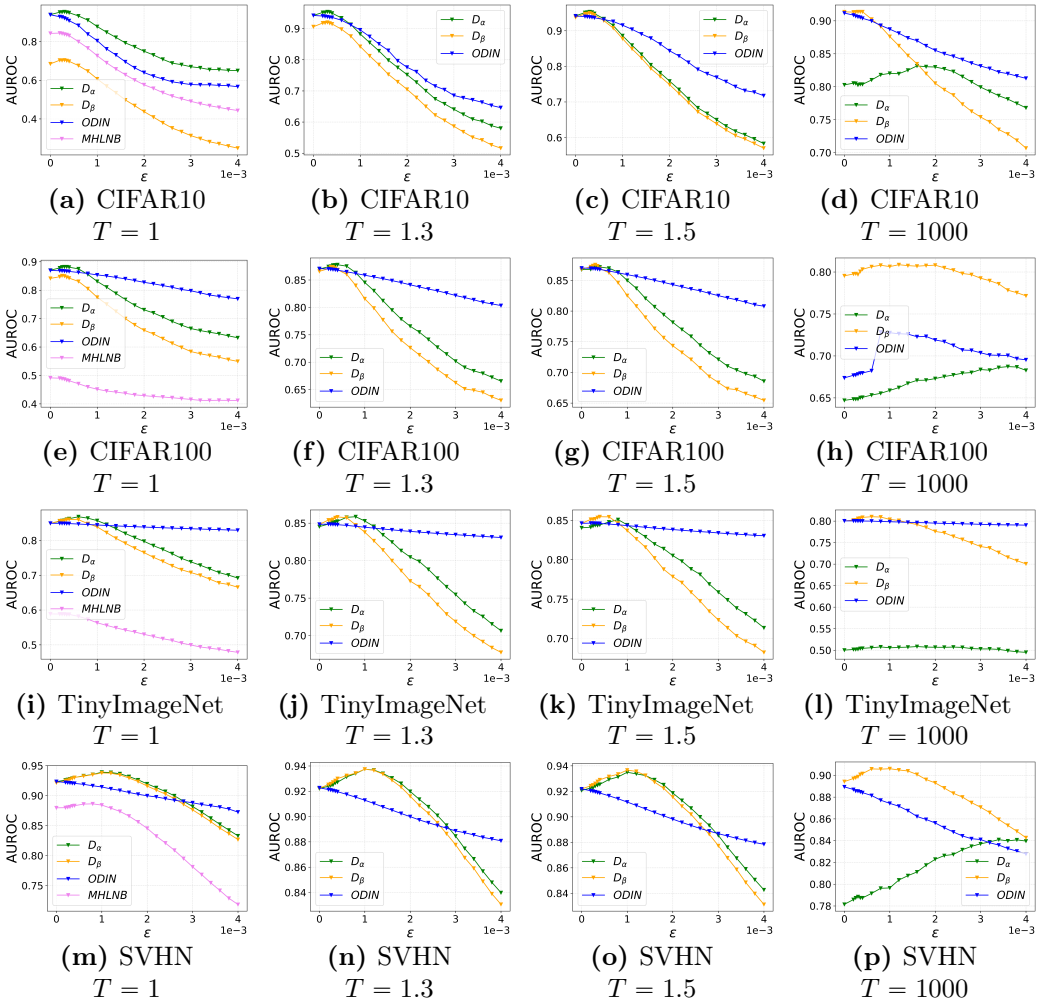
Figure A.2: The effect of varying T and ϵ. Comparison of AUROCs obtained via D_α (in green) and via D_β (in orange) for different values of T and ϵ.

Comparison D_α and D_β

We include the plots for DOCTOR*: comparison between D_α and D_β* (Section 3.4.3). In Fig. A.2a, Fig. A.2d, Fig. A.2g and Fig. A.2j, we set ϵ at its best value which is found to coincide in the case of D_α and D_β. In Fig. A.2b, Fig. A.2e, Fig. A.2h and Fig. A.2k we do the opposite and we set T to its best value w.r.t. D_α whilst in Fig. A.2c, Fig. A.2f, Fig. A.2i and Fig. A.2l, the value of T is chosen w.r.t. the best value for D_β.

Comparison D_α, D_β, ODIN and MHLNB

We conclude by showing in Fig. A.3 the test results obtained by varying T and ϵ in PBB for all the methods. We present 4 groups of plots (one for each image dataset) and in each plot we pick T from $\{1, 1.3, 1.5, 1000\}$ (the values selected for D_α, D_β, ODIN and MHLNB Table 3.1) and we let ϵ vary.



(a) CIFAR10 $T = 1$ **(b)** CIFAR10 $T = 1.3$ **(c)** CIFAR10 $T = 1.5$ **(d)** CIFAR10 $T = 1000$

(e) CIFAR100 $T = 1$ **(f)** CIFAR100 $T = 1.3$ **(g)** CIFAR100 $T = 1.5$ **(h)** CIFAR100 $T = 1000$

(i) TinyImageNet $T = 1$ **(j)** TinyImageNet $T = 1.3$ **(k)** TinyImageNet $T = 1.5$ **(l)** TinyImageNet $T = 1000$

(m) SVHN $T = 1$ **(n)** SVHN $T = 1.3$ **(o)** SVHN $T = 1.5$ **(p)** SVHN $T = 1000$

Figure A.3: Overall results when varying T and ϵ.

Misclassification detection in presence of out-of-distribution samples

We include in Table A.7 the results of all the simulations carried out for detecting misclassification detection in presence of out-of-distribution samples. The experimental setting is reported in Section 3.4.2.

A.3.5 DOCTOR for pure OOD detection

It is worth emphasizing that DOCTOR is not targeting OOD detection, which is a rather different problem from the one investigated in this paper. So we did not optimize an ad-hoc input perturbation for DOCTOR within the OOD detection setup, i.e. we kept the same input perturbation proposed for the misclassification

detection task. The baseline results reported in Table A.4 show that DOCTOR is competitive for OOD detection as well since it can reach similar scores or even outperform the baseline (e.g., the simulations with LSUN (CROP) show an improvement of the results of 3.3% in terms of FRR %). We indicate the methods together with their parameter setting. ODIN_{OOD} denotes the same parameter setting as in [LLS18].

Table A.4: DOCTOR for pure OOD detection. We set : $\epsilon_\alpha = 0$ and $T_\alpha = 15$, $\epsilon_\beta = 0$ and $T_\beta = 1000$, as in [LLS18] for ODIN_{OOD}. The baseline results reported below show that DOCTOR is competitive for OOD detection as well since it can reach similar scores or even outperform the baseline.

DATASET-IN	DATASET-OUT	AUROC %			FRR % (95 % TRR)		
		D_α	D_β	ODIN_{OOD}	D_α	D_β	ODIN_{OOD}
CIFAR10	iSUN	98.1	97.9	**98.8**	8	9.1	**6.3**
	TINY (RES)	97.6	97.3	**98.5**	9.9	11.2	**7.2**
	LSUN (CROP)	**98.6**	98.2	98.2	**5.4**	6.9	8.7
	TINY (CROP)	98.9	98.5	**99.1**	4.6	6.4	**4.3**

DOCTOR in presence of OOD samples that are similar to in-distribution ones

We tested DOCTOR in pure OOD setting, considering CIFAR100 as in-distribution and CIFAR10 as out-distribution. The results below show that DOCTOR optimized as in the following paper outperforms ODIN (optimized as described in [LLS18]) and ENERGY. This is particularly promising as it shows that DOCTOR, without performing any training and without been particularly optimized for OOD detection, can perform well on a wider variety of problems.

A.3.6 Some observations on the white-box scenario (WB)

It is worth clarifying the results in Table A.6 to motivate the performance obtained using the Mahalanobis-based discriminator (MHLNB - WB) for the misclassification detection problem and the issues it raises. First of all, we emphasize that given a network and an input sample DOCTOR only needs to access the logits

Table A.5: DOCTOR for OOD detection of samples similar to the in-distribution ones. Comparison of D_α with ENERGY and ODIN (parameter setting as in [LLS18]).

DATASET-IN	DATASET-OUT	METHODS	AUROC %	FRR % (95 % TRR)
CIFAR100	CIFAR10	D_α (PBB)	76.8	64.2
		ENERGY	73.3	76.4
		ODIN (OOD)	70.5	79.5

Table A.6: White-box setting. Comparison of MHLNB (WB) and D_α (PBB).

DATASET-IN	METHODS	AUROC %	FRR % (95 % TRR)
CIFAR10	D_α (PBB)	95.2	13.9
	MHLNB (WB)	49.5	97.3
CIFAR100	D_α (PBB)	88.2	35.7
	MHLNB (WB)	51.6	94.9

output of the network in order to perform the detection. On the contrary, the detector based on Mahalanobis distance consists of 3 steps:

- Estimation of the class mean and covariance matrix;
- Features extraction according to the Mahalanobis score function;
- Aggregation of the scores obtained layer by layer in order to obtain a decision a rule for the discriminator.

Clearly, the Mahalanobis distance-based method requires additional samples compared to DOCTOR. Although estimating the mean and the covariance matrix is possible by exploiting samples from the benchmark training set (e.g. CIFAR10, CIFAR100, ...), this method still needs additional (different from training) samples for learning the linear regressor intended to distinguish between correctly (positive) and incorrectly (negative) classified samples. In order to generate the negative samples, we consider the use of adversarial examples generated through Projected Gradient Descent Attack (magnitude of the perturbation 0.0031), which does not assume any knowledge about the test set.

Table A.7: Misclassification detection in presence of OOD samples: overall results. In PBB we set $\epsilon_\alpha = 0.00035$ and $T_\alpha = 1$, $\epsilon_\beta = 0.00035$ and $T_\beta = 1.5$, $\epsilon_{\text{ODIN}} = 0$ and $T_{\text{ODIN}} = 1.3$. By ODIN_{ood}, we mean ODIN with the parameter setting as in [LLS18]. Since we proceed in a Monte Carlo fashion, the results are reported in terms of *mean / standard deviation*. In TBB for by ODIN we report the results of SR, since both methods coincide when $T = 1$ and $\epsilon = 0$.

DATASET (IN)	DATASET (OUT)	SCENARIO	AUROC %				FRR % (95 % TRR)			
			D_α	D_β	ODIN	ODIN_{OOD}	D_α	D_β	ODIN	ODIN_{OOD}
CIFAR10 ♣	iSUN	PBB	**95.4** / 0.1	95.1 / 0.1	94.6 / 0.1	89.6 / 0	14 / 0.5	**13.5** / 0.4	17.2 / 0.3	38.9 / 0
		TBB	**94.6** / 0	69.3 / 0.1	94.5 / 0.1	-	**17.7** / 0.1	**17.7** / 0.1	**17.7** / 0	-
	LSUN (CROP)	PBB	**95.5** / 0.1	95.1 / 0	94.7 / 0	92.6 / 0	13.1 / 0.5	**13** / 0.2	17.3 / 0	31.9 / 0.1
		TBB	**94.4** / 0.1	69.2 / 0.1	**94.4** / 0	-	**17.6** / 0.2	**17.6** / 0.2	17.7 / 0.2	-
	LSUN (RESIZE)	PBB	**95.4** / 0.1	95.1 / 0	94.8 / 0	89.6 / 0	13.4 / 0.6	**13.2** / 0.3	17 / 0.3	38.9 / 0
		TBB	**94.6** / 0.1	69.3 / 0.1	94.5 / 0.1	-	**17.8** / 0.1	**17.8** / 0.1	**17.8** / 0.1	-
	TINY (CROP)	PBB	**95.4** / 0	95.1 / 0.1	94.7 / 0	89.6 / 0	13.4 / 0.4	**13** / 0.2	17.2 / 0.3	38.9 / 0
		TBB	**94.6** / 0	69.4 / 0.1	**94.6** / 0	-	**17.8** / 0.1	**17.8** / 0.1	**17.8** / 0.1	-
	TINY (RES)	PBB	**95.2** / 0.1	94.9 / 0	94.6 / 0.1	89.6 / 0	**14** / 0.4	**14** / 0.5	17.8 / 0.4	38.9 / 0
		TBB	**94.4** / 0.1	69.2 / 0	**94.4** / 0	-	**17.8** / 0.1	**17.8** / 0.1	**17.8** / 0.1	-
CIFAR100 ♣	iSUN	PBB	**86.5** / 0.2	85.8 / 0	85.6 / 0.2	79 / 0.1	**45.3** / 1	46.1 / 0.5	46.8 / 1	65.9 / 0.4
		TBB	**85.6** / 0.1	82.7 / 0.1	85.5 / 0.1	-	46.9 / 0.4	**46.8** / 0.4	**46.8** / 0.4	-
	LSUN (CROP)	PBB	**89.1** / 0	88.5 / 0.1	88 / 0.1	80.6 / 0	**35.6** / 0.4	35.7 / 0.2	39.9 / 0.3	65.1 / 0
		TBB	**87.9** / 0.1	84.9 / 0.1	87.7 / 0.1	-	**39.8** / 0.6	**39.8** / 0.6	**39.8** / 0.6	-
	LSUN (RESIZE)	PBB	**86.8** / 0.1	86.2 / 0.1	86 / 0.1	79.1 / 0.1	**44.4** / 0.9	**44.4** / 0.6	45.3 / 0.3	65.4 / 0.3
		TBB	**85.8** / 0.1	82.9 / 0.1	85.7 / 0.1	-	45.9 / 0.5	**45.8** / 0.5	**45.8** / 0.5	-
	TINY (CROP)	PBB	**88.4** / 0.1	87.8 / 0.1	87.6 / 0.1	81.8 / 0.1	38.2 / 0.4	**37.8** / 0.9	40.6 / 0.5	63.4 / 0.1
		TBB	**87.2** / 0.1	84.2 / 0.1	87 / 0.1	-	**42** / 0.6	**42** / 0.6	**42** / 0.6	-
	TINY (RES)	PBB	**86.8** / 0.1	86.3 / 0.1	85.9 / 0.1	79.2 / 0.1	44 / 0.1	**43.6** / 0.2	45.9 / 1.2	65.8 / 0.3
		TBB	**85.9** / 0.2	83 / 0.2	85.8 / 0.2	85.8 / 0.2	**45.7** / 1.3	**45.7** / 1.3	**45.7** / 1.3	-
CIFAR10 ♢	iSUN	PBB	**95.5** / 0.1	95.3 / 0.1	94.9 / 0.1	91.5 / 0	14.4 / 0.6	**13.4** / 0.2	16.8 / 0.5	34/ 0.1
		TBB	**95** / 0	69.6 / 0	94.9 / 0.1	-	**16.4** / 0.2	**16.4** / 0.2	**16.4** / 0.2	-
	LSUN (CROP)	PBB	**95.8** / 0.1	95.5 / 0.1	95 / 0.1	93.9 / 0.1	**12.4** / 0.2	12.6 / 0.1	16.1 / 0.4	24.8 / 0.1
		TBB	**94.8** / 0.1	69.6 / 0.1	**94.8** / 0.1	-	16.7 / 0.4	16.8 / 0.4	**16.6** / 0.4	-
	LSUN (RESIZE)	PBB	**95.8** / 0	95.6 / 0	95.2 / 0	91.6 / 0	**12.9** / 0.5	**12.9** / 0.3	15.8 / 0.2	33.9 / 0
		TBB	**95** / 0	69.7 / 0.1	**95** / 0.1	-	**16.4** / 0.2	**16.4** / 0.3	**16.4** / 0.2	-
	TINY (CROP)	PBB	**95.8** / 0.1	95.5 / 0.1	95.2 / 0.1	91.5 / 0	**12.8** / 0.7	12.9 / 0.5	16 / 0	33.9 / 0
		TBB	**95** / 0.2	69.8 / 0.1	**95** / 0.1	-	**16.4** / 0.2	16.5 / 0.2	**16.4** / 0.2	-
	TINY (RES)	PBB	**95.4** / 0.1	95 / 0.1	94.8 / 0.1	91.4 / 0	15 / 0.1	**14.8** / 0.7	17 / 0.5	34.5 / 0.9
		TBB	**94.6** / 0.2	69.3 / 0.2	**94.6** / 0.2	-	18.1 / 1	18.1 / 1.1	**18** / 1	-
CIFAR100 ♢	iSUN	PBB	**84.8** / 0.1	84.4 / 0.2	84.6 / 0.1	80.8 / 0.2	53.6 / 1	**51.2** / 0.2	51.3 / 0.1	63.5 / 0.3
		TBB	**84.1** / 0.1	81.2 / 0.1	84 / 0.1	-	**52.5** / 0.5	**52.5** / 0.5	**52.5** / 0.5	-
	LSUN (CROP)	PBB	**89.9** / 0.1	89.6 / 0	89 / 0	84.1 / 0	**35.2** / 0.7	35.4 / 0.2	39.3 / 0.1	62.2 / 0
		TBB	**88.7** / 0.1	85.7 / 0	88.5 / 0.1	-	**38.8** / 0.5	**38.8** / 0.5	**38.8** / 0.4	-
	LSUN (RESIZE)	PBB	**85.3** / 0.3	85.1 / 0.2	84.9 / 0.1	81.1 / 0	51.6 / 0.9	**48.8** / 1	49.2 / 0.7	63.3 / 0.1
		TBB	**84.6** / 0.2	81.8 / 0.2	**84.6** / 0.1	-	**50.6** / 0.8	50.7 / 0.8	**50.6** / 0.8	-
	TINY (CROP)	PBB	**88.2** / 0	88.1 / 0.2	87.7 / 0.1	84.8 / 0.1	41.2 / 0.3	**40.2** / 0.6	42.3 / 0.4	59/ 0.2
		TBB	**87.7** / 0.1	84.7 / 0.1	87.5 / 0.1	-	**41.8** / 0.5	**41.8** / 0.5	**41.8** / 0.5	-
	TINY (RES)	PBB	**85.4** / 0.2	84.8 / 0.2	85.1 / 0.3	81.2 / 0.1	51.8 / 1.6	52 / 0.8	**50.4** / 0.9	63.3 / 0.2
		TBB	**84.8** / 0.1	81.9 / 0.1	84.7 / 0.1	-	**51.4** / 0.5	**51.4** / 0.5	**51.4** / 0.5	-
CIFAR10 ♠	iSUN	PBB	**95.6** / 0.1	**95.6** / 0	95.4 / 0	93.5 / 0	15.1 / 0.1	**13.6** / 0.5	16.1 / 0.2	30.6 / 0.4
		TBB	**95.4** / 0.1	70 / 0.1	95.2 / 0.1	-	16.1 / 0.4	**16** / 0.5	**16** / 0.4	-
	LSUN (CROP)	PBB	**96.1** / 0.1	95.9 / 0.1	95.5 / 0.2	95.2 / 0.1	12.6 / 0.5	**12.4** / 0.3	15.3 / 0.7	20.8 / 0.4
		TBB	**95.2** / 0.1	70 / 0.1	**95.2** / 0.1	-	15.8 / 0.7	15.8 / 0.7	**15.7** / 0.7	-
	LSUN (RESIZE)	PBB	**96** / 0	95.8 / 0	95.7 / 0	93.6 / 0	13.2 / 0.5	**13** / 0.2	15.2 / 0.4	30.3 / 0.4
		TBB	**95.5** / 0.1	70.2 / 0.1	**95.5**/ 0.1	-	15.2 / 0.5	15.2 / 0.5	**15.1** / 0.5	-
	TINY (CROP)	PBB	**96** / 0.1	95.9 / 0.1	95.7 / 0	93.6 / 0	13.5 / 0.9	**12.7** / 0.4	15.2 / 0.4	30.3 / 0.4
		TBB	95.5 / 0.1	70.3 / 0	**95.6** / 0	-	15.1 / 0.2	**15** / 0.3	**15** / 0.2	-
	TINY (RES)	PBB	**95.5** / 0.1	95.2 / 0.1	95.1 / 0.1	93.2	**14.7** / 0.3	14.8 / 0.5	17.1 / 0.4	31/ 0
		TBB	**94.9** / 0.1	69.7 / 0.1	**94.9** / 0.1	-	16.8 / 0.3	16.9 / 0.2	**16.7** / 0.2	-
CIFAR100 ♠	iSUN	PBB	**83.3** / 0.1	83.1 / 0.1	83 / 0.2	82.6 / 0.2	57.8 / 0.3	57.1 / 1	**56.8** / 0.8	60/ 0.4
		TBB	**82.6** / 0.2	79.7 / 0.2	82.5 / 0.2	-	**58.3** / 1	58.4 / 1.1	58.4 / 1	-
	LSUN (CROP)	PBB	90.6 / 0	**90.7** / 0	89.9 / 0.1	87.5 / 0	35.9 / 0.2	**34.6** / 0.2	38.5 / 0.4	56.1 / 0.2
		TBB	**89.4** / 0.1	86.2 / 0	89 / 0	-	**39.4** / 0.1	**39.4** / 0.1	**39.4** / 0.1	-
	LSUN (RESIZE)	PBB	83.6 / 0.2	**83.8** / 0.1	83.6 / 0.2	83.2 / 0.1	55.8 / 0.4	54.2 / 0.7	**54.1** / 0.6	59.6 / 0.8
		TBB	**83.2** / 0.1	80.4 / 0.1	**83.2** / 0.1	-	**55** / 0.6	**55** / 0.7	**55** / 0.6	-
	TINY (CROP)	PBB	88.3 / 0.1	**88.5** / 0.1	88.1 / 0.1	87.7 / 0.1	43.2 / 0.5	**41.5** / 0.7	42.9 / 0.4	54.3 / 0.1
		TBB	**87.8** / 0	84.7 / 0.1	87.5 / 0.1	-	**43.7** / 0.2	**43.7** / 0.2	**43.7** / 0.2	-
	TINY (RES)	PBB	83.8 / 0.1	83.8 / 0.1	**83.9** / 0.2	83/ 0.2	57.9 / 0.5	56.6 / 0.9	**55.6** / 1	61/ 0.6
		TBB	**83.6** / 0.1	80.7 / 0.1	83.5 / 0.1	-	**55.5** / 0.8	**55.5** / 0.8	**55.5** / 0.8	-

APPENDIX B

Appendix to Chapter 4

B.1 Additional results

In the following sections, we include results that, due to space limitations, were not included in Section 4.4.2.

B.1.1 Additional Results on CIFAR10

Attack Rate

In Table B.1, we report the average number of successful attacks per natural sample considered in MEAD and in the single-armed settings.

As explained in Section 4.4.2, the attacks generated thanks to the Adversarial Cross-Entropy seem to be the most harmful ones for the underlying classifier. It is not surprising given that the Cross-Entropy was used as the loss to train the classifier. Attacks generated through the maximization of the Kullback-Leibler divergence, the Gini Impurity score, and the Fisher-Rao distance all seem to be equally damaging.

NSS

In Table B.2, we report the performances of the NSS detector in CIFAR10.

NSS is, by far, the best performing method to detect adversarial examples under the MEAD framework that we consider in this paper. The decrease in performances due to the worst-case scenario that we consider is up to 5.0 percentage points in terms of AUROC↑%. Under the single-armed setting, *NSS* is the most

Avg. Num. of Successful Attack / Tot. Num. of Attack					
Norm L_1	MEAD	ACE	KL	Gini	FR
PGD1					
$\varepsilon = 5$	1.28 / 4	0.45 / 1	0.31 / 1	0.27 / 1	0.25 / 1
$\varepsilon = 10$	2.24 / 4	0.81 / 1	0.53 / 1	0.47 / 1	0.43 / 1
$\varepsilon = 15$	2.60 / 4	0.92 / 1	0.60 / 1	0.59 / 1	0.50 / 1
$\varepsilon = 20$	2.72 / 4	0.95 / 1	0.61 / 1	0.64 / 1	0.52 /1
$\varepsilon = 25$	2.76 / 4	0.95 / 1	0.61 / 1	0.67 / 1	0.53 / 1
$\varepsilon = 30$	2.80 / 4	0.95 / 1	0.62 / 1	0.68 / 1	0.54 / 1
$\varepsilon = 40$	2.80 / 4	0.95 / 1	0.62 / 1	0.70 / 1	0.54 / 1
Norm L_2					
PGD2					
$\varepsilon = 0.125$	1.12 / 4	0.39 / 1	0.27 / 1	0.24 / 1	0.22 / 1
$\varepsilon = 0.25$	2.16 / 4	0.78 / 1	0.51 / 1	0.45 / 1	0.40 / 1
$\varepsilon = 0.3125$	2.40 / 4	0.86 / 1	0.56 / 1	0.54 / 1	0.45 / 1
$\varepsilon = 0.5$	2.68 / 4	0.93 / 1	0.60 / 1	0.64 / 1	0.50 / 1
$\varepsilon = 1$	2.76 / 4	0.94 / 1	0.61 / 1	0.70 / 1	0.52 / 1
$\varepsilon = 1.5$	2.76 / 4	0.94 / 1	0.61 / 1	0.70 / 1	0.52 / 1
$\varepsilon = 2$	2.76 / 4	0.94 / 1	0.61 / 1	0.70 / 1	0.52 / 1
DeepFool					
No ε	0.48 / 1	0.48 / 1	0.48 / 1	0.48 / 1	0.48 / 1
CW2					
$\varepsilon = 0.01$	0.95 / 1	0.95 / 1	0.95 / 1	0.95 / 1	0.95 / 1
HOP					
$\varepsilon = 0.1$	0.41 / 1	0.41 / 1	0.41 / 1	0.41 / 1	0.41 / 1
Norm L_∞					
PGDi, FGSM, BIM					
$\varepsilon = 0.03125$	8.28 / 12	2.76 / 3	1.86 / 3	2.13 / 3	1.53 / 3
$\varepsilon = 0.0625$	8.52 / 12	2.85 / 3	1.89 / 3	2.22 / 3	1.56 / 3
$\varepsilon = 0.25$	8.88 / 12	2.85 / 3	1.98 / 3	2.31 / 3	1.71 / 3
$\varepsilon = 0.5$	9.24 / 12	2.85 / 3	2.13 / 3	2.31 / 3	1.92 / 3
PGDi, FGSM, BIM, SA					
$\varepsilon = 0.125$	9.00 / 13	3.21 / 4	2.26 / 4	2.61 / 4	1.95 / 4
PGDi, FGSM, BIM, CWi					
$\varepsilon = 0.3125$	9.88 / 13	3.79 / 4	2.99 / 4	3.25 / 4	2.73 / 4
No norm					
STA					
No ε	0.24 / 1	0.24 / 1	0.24 / 1	0.24 / 1	0.24 / 1

Table B.1: Average number of successful attacks per natural sample considered in the single-armed setting and MEAD (CIFAR10). The results are reported in the table together with the total number of attacks performed per natural sample(*Avg. / Tot.*). No norm denotes the group of attacks that do not depend on the norm constraint.

sensitive to the Kullback-Leibler divergence, even though the results are quite similar amongst the different attack objectives.

NSS	MEAD		ACE		KL		Gini		FR	
Norm L_1	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD1										
$\varepsilon = 5$	**48.5**	**94.2**	49.9	93.5	49.6	93.0	50.3	93.2	49.9	93.3
$\varepsilon = 10$	**54.0**	**90.3**	56.9	88.4	56.6	88.3	57.1	88.8	57.0	88.1
$\varepsilon = 15$	**58.8**	**86.8**	63.1	83.0	62.8	83.1	63.5	84.0	63.2	82.5
$\varepsilon = 20$	**63.5**	**82.3**	68.5	77.1	68.1	77.3	69.9	77.6	68.7	76.4
$\varepsilon = 25$	**67.7**	**77.2**	73.1	71.1	72.7	71.8	75.0	71.4	73.4	70.9
$\varepsilon = 30$	**71.4**	**73.4**	77.1	64.5	76.8	65.1	78.6	67.3	77.4	65.2
$\varepsilon = 40$	**76.1**	**67.3**	83.5	52.7	83.3	53.5	80.1	64.9	83.6	52.7
L_1 Average	**62.9**	**81.6**	67.4	75.7	67.1	76.0	67.8	78.2	67.6	75.6
Norm L_2	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD2										
$\varepsilon = 0.125$	**48.3**	**94.3**	49.5	93.8	49.1	93.5	49.5	**94.3**	49.6	93.5
$\varepsilon = 0.25$	**53.2**	**91.2**	55.9	89.1	55.6	89.2	55.9	89.8	55.8	89.4
$\varepsilon = 0.3125$	**55.8**	**89.2**	59.4	86.5	59.0	86.6	59.3	87.7	59.3	86.6
$\varepsilon = 0.5$	**63.3**	**82.6**	68.3	77.4	68.0	77.4	69.0	78.7	68.4	77.2
$\varepsilon = 1$	**76.4**	**67.5**	84.4	50.6	84.3	50.5	79.3	66.8	84.7	50.7
$\varepsilon = 1.5$	81.0	63.0	92.8	28.7	92.7	28.9	**79.5**	**66.5**	93.0	27.3
$\varepsilon = 2$	82.6	62.3	96.8	13.9	96.9	13.1	**79.5**	**66.5**	95.9	17.2
DeepFool										
No ε	**57.0**	**91.7**	**57.0**	**91.7**	**57.0**	**91.7**	**57.0**	**91.7**	**57.0**	**91.7**
CW2										
$\varepsilon = 0.01$	**56.4**	**90.8**	**56.4**	**90.8**	**56.4**	**90.8**	**56.4**	**90.8**	**56.4**	**90.8**
HOP										
$\varepsilon = 0.1$	**66.1**	**87.0**	**66.1**	**87.0**	**66.1**	**87.0**	**66.1**	**87.0**	**66.1**	**87.0**
L_2 Average	**64.0**	**82.0**	68.7	71.0	68.5	70.9	65.1	82.0	68.6	71.1
Norm L_∞	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGDi, FGSM, BIM										
$\varepsilon = 0.03125$	**83.0**	**55.3**	88.5	42.3	89.6	39.9	87.5	47.8	89.3	39.8
$\varepsilon = 0.0625$	**96.0**	**17.2**	98.1	7.9	98.4	6.8	97.1	13.2	98.4	6.8
$\varepsilon = 0.25$	**97.3**	**0.6**	99.7	**0.6**	99.7	**0.6**	97.8	**0.6**	98.0	**0.6**
$\varepsilon = 0.5$	**82.5**	**100.0**	99.7	0.6	99.7	0.6	86.2	**100.0**	85.7	**100.0**
PGDi, FGSM, BIM, SA										
$\varepsilon = 0.125$	9.4	99.9	**9.4**	**99.9**	**9.4**	**99.9**	**9.4**	**99.9**	**9.4**	**99.9**
PGDi, FGSM, BIM, CWi										
$\varepsilon = 0.3125$	**63.2**	**99.1**	66.1	89.4	66.1	89.4	63.9	96.2	63.9	95.8
L_∞ Average	**71.9**	**62.0**	76.9	40.1	77.2	39.5	73.7	59.6	74.1	57.2
No norm	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
STA										
No ε	**88.5**	**38.8**	**88.5**	**38.8**	**88.5**	**38.8**	**88.5**	**38.8**	**88.5**	**38.8**
No norm Average	**88.5**	**38.8**	**88.5**	**38.8**	**88.5**	**38.8**	**88.5**	**38.8**	**88.5**	**38.8**

Table B.2: Performances on *NSS* per objective and in MEAD on CIFAR10. The worst results among all the settings is in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

KD-BU

In Table B.3, we show the result of our MEAD framework as well as the single-armed settings on CIFAR10, evaluated on *KD-BU*.

KD-BU seems to work poorly on MEAD as well as on the single-armed setting. For most settings, *KD-BU* is worst than a random detector. The decrease in AUROC↑% between the worst single-armed setting and MEAD is up to 24.9 percentage points.

KD-BU	MEAD		ACE		KL		Gini		FR	
Norm L_1	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
PGD1										
$\varepsilon = 5$	**41.3**	**96.6**	58.0	94.7	57.1	94.8	67.9	93.0	56.5	94.8
$\varepsilon = 10$	**36.9**	**97.2**	55.1	94.7	54.0	94.7	72.4	91.9	54.6	94.8
$\varepsilon = 15$	**39.9**	**96.7**	57.9	94.5	57.9	94.5	73.2	92.9	58.2	94.5
$\varepsilon = 20$	**47.3**	**96.3**	66.7	93.2	66.9	93.2	75.9	92.3	65.9	93.4
$\varepsilon = 25$	**55.5**	**95.6**	75.5	91.0	76.2	90.8	76.5	92.0	75.7	91.0
$\varepsilon = 30$	**62.6**	**94.7**	83.5	86.9	84.3	86.3	77.2	91.8	84.1	86.4
$\varepsilon = 40$	**72.6**	**92.6**	93.5	65.4	93.5	64.5	77.0	91.9	93.7	63.9
L_1 Average	**50.9**	**95.7**	70.0	88.6	70.0	88.4	74.3	92.3	69.8	88.4
Norm L_2	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
PGD2										
$\varepsilon = 0.125$	**42.0**	**96.6**	59.0	94.6	58.2	94.6	68.2	92.6	57.8	94.6
$\varepsilon = 0.25$	**38.4**	**96.8**	54.2	95.0	53.9	95.0	70.5	92.5	55.0	94.8
$\varepsilon = 0.3125$	**38.6**	**96.8**	55.1	94.7	55.8	94.6	72.9	92.0	55.6	94.6
$\varepsilon = 0.5$	**47.9**	**96.2**	66.8	93.2	67.8	92.9	75.4	92.4	67.0	93.1
$\varepsilon = 1$	**74.2**	**92.1**	94.4	58.1	94.6	55.7	77.1	91.8	94.7	57.3
$\varepsilon = 1.5$	80.1	90.1	99.0	0.0	99.2	0.0	**77.1**	**91.9**	99.3	0.0
$\varepsilon = 2$	81.3	89.7	99.8	0.0	99.9	0.0	**77.1**	**91.9**	99.8	0.0
DeepFool										
No ε	**67.1**	**94.0**	**67.1**	**94.0**	**67.1**	**94.0**	**67.1**	**94.0**	**67.1**	**94.0**
CW2										
$\varepsilon = 0.01$	**53.0**	**95.1**	**53.0**	**95.1**	**53.0**	**95.1**	**53.0**	**95.1**	**53.0**	**95.1**
HOP										
$\varepsilon = 0.1$	**67.3**	**94.0**	**67.3**	**94.0**	**67.3**	**94.0**	**67.3**	**94.0**	**67.3**	**94.0**
L_2 Average	**59.0**	**94.1**	71.6	71.9	71.7	71.6	70.6	92.8	71.7	71.8
Norm L_∞	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
PGDi, FGSM, BIM										
$\varepsilon = 0.03125$	**29.2**	**97.3**	52.8	94.7	62.4	92.7	46.5	96.2	58.2	93.8
$\varepsilon = 0.0625$	**35.2**	**96.9**	67.1	91.7	74.8	88.7	52.3	95.7	75.0	89.0
$\varepsilon = 0.25$	**45.1**	**96.4**	84.4	84.8	83.5	86.6	65.0	94.5	81.8	88.5
$\varepsilon = 0.5$	**43.1**	**96.6**	77.4	90.7	79.5	89.2	68.0	94.1	83.4	88.3
PGDi, FGSM, BIM, SA										
$\varepsilon = 0.125$	**34.9**	**97.0**	59.2	94.9	60.5	94.9	46.5	96.4	60.6	94.9
PGDi, FGSM, BIM, CWi										
$\varepsilon = 0.3125$	**33.2**	**97.1**	47.8	95.8	47.7	95.8	43.8	96.4	47.7	95.9
L_∞ Average	**36.8**	**96.9**	64.8	92.1	68.1	91.3	53.7	95.6	67.8	91.7
No norm	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
STA										
No ε	**65.4**	**94.2**	**65.4**	**94.2**	**65.4**	**94.2**	**65.4**	**94.2**	**65.4**	**94.2**
No norm Average	**65.4**	**94.2**	**65.4**	**94.2**	**65.4**	**94.2**	**65.4**	**94.2**	**65.4**	**94.2**

Table B.3: Performances on *KD-BU* per objective and in MEAD on CIFAR10. The worst results among all the settings are shown in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

LID

In Table B.4, we report the detection performances of the *LID* method under the MEAD framework and the different single-armed settings.

LID is quite sensitive to all the attacker's objectives. Depending on the norm-constraint and on the ε value, each one of the four objectives can be the most harmful one. Moreover, this detection method is quite affected by the MEAD setting. Indeed, the decrease of performances in terms of AUROC↑%due to the use of the worst-case scenario is up to 28.9 percentage points compared to the worst single-armed setting (value obtained under the L_1-norm constraint for $\varepsilon = 40$).

LID	MEAD		ACE		KL		Gini		FR	
Norm L_1	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
PGD1										
$\varepsilon = 5$	**57.5**	**93.5**	66.2	86.9	65.5	88.0	77.9	77.0	64.4	88.7
$\varepsilon = 10$	**48.3**	**96.4**	62.3	89.6	60.6	92.0	84.3	70.7	61.9	90.2
$\varepsilon = 15$	**53.1**	**95.4**	61.6	90.7	61.8	90.9	89.1	55.6	61.8	91.1
$\varepsilon = 20$	**37.5**	**99.0**	65.7	89.2	65.4	90.2	91.4	40.8	66.3	87.1
$\varepsilon = 25$	**47.7**	**96.3**	70.7	84.0	70.9	84.7	92.8	37.1	70.3	83.3
$\varepsilon = 30$	**56.4**	**95.0**	76.1	75.7	76.5	79.8	93.2	35.2	75.5	81.9
$\varepsilon = 40$	**54.9**	**92.5**	84.6	58.5	85.0	54.7	93.3	32.6	83.8	63.3
L_1 Average	**50.8**	**95.4**	69.6	82.1	69.4	82.9	88.9	49.9	69.1	83.7
Norm L_2	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
PGD2										
$\varepsilon = 0.125$	**61.3**	**90.6**	67.5	86.3	66.2	88.7	77.8	76.3	64.6	88.8
$\varepsilon = 0.25$	**47.9**	**96.5**	62.0	90.3	60.8	91.5	84.1	68.3	61.4	90.6
$\varepsilon = 0.3125$	**49.7**	**95.8**	60.8	91.5	60.5	91.5	87.4	60.3	61.9	88.8
$\varepsilon = 0.5$	**47.6**	**97.9**	66.1	87.7	65.6	91.2	91.0	49.2	65.9	88.9
$\varepsilon = 1$	**65.2**	**84.1**	86.0	50.6	86.3	48.9	93.4	32.5	85.2	55.9
$\varepsilon = 1.5$	**77.0**	**56.2**	94.0	20.0	93.9	19.8	93.5	31.5	93.1	21.7
$\varepsilon = 2$	**81.5**	**46.4**	96.3	11.9	96.3	12.2	93.4	31.8	95.0	15.2
DeepFool										
No ε	**70.9**	**86.9**	**70.9**	**86.9**	**70.9**	**86.9**	**70.9**	**86.9**	**70.9**	**86.9**
CW2										
$\varepsilon = 0.01$	**61.6**	**92.5**	**61.6**	**92.5**	**61.6**	**92.5**	**61.6**	**92.5**	**61.6**	**92.5**
HOP										
$\varepsilon = 0.1$	**72.2**	**83.6**	**72.2**	**83.6**	**72.2**	**83.6**	**72.2**	**83.6**	**72.2**	**83.6**
L_2 Average	**63.5**	**83.1**	73.7	70.1	73.4	70.7	82.5	61.3	73.2	71.3
Norm L_∞	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
PGDi, FGSM, BIM										
$\varepsilon = 0.03125$	**41.4**	**96.5**	55.5	89.9	60.5	88.9	69.7	86.7	65.5	82.5
$\varepsilon = 0.0625$	**58.4**	**83.9**	76.7	59.0	78.6	57.0	83.6	59.2	87.1	44.8
$\varepsilon = 0.25$	**58.0**	**89.5**	88.4	27.0	92.1	41.7	77.7	70.9	93.0	20.6
$\varepsilon = 0.5$	**58.6**	**86.7**	90.3	25.7	93.6	19.2	76.3	74.3	91.8	23.9
PGDi, FGSM, BIM, SA										
$\varepsilon = 0.125$	**56.7**	**91.8**	82.5	48.6	85.6	49.8	64.6	91.0	85.5	48.8
PGDi, FGSM, BIM, CWi										
$\varepsilon = 0.3125$	**49.5**	**96.1**	60.5	90.7	68.7	88.8	56.0	96.0	68.8	87.9
L_∞ Average	**53.8**	**90.8**	75.7	56.8	79.9	57.6	71.3	79.7	82.0	51.4
No norm	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%	AUROC↑%%	FPR$_{\downarrow 95\%}$%%
STA										
No ε	**88.0**	**58.1**	**88.0**	**58.1**	**88.0**	**58.1**	**88.0**	**58.1**	**88.0**	**58.1**
No norm Average	**88.0**	**58.1**	**88.0**	**58.1**	**88.0**	**58.1**	**88.0**	**58.1**	**88.0**	**58.1**

Table B.4: Performances on *LID* per objective and in MEAD on CIFAR10. The worst results among all the settings is in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

FS

In Table B.5, we present the summary of the *FS* detection method.

FS is not quite affected by the MEAD framework under the L_1 and L_2-norm constraints. The reason why is explained in Section 4.4.2 in the remark of the paragraph called **MEAD and the single-armed setting**. However, under the L_∞-norm constraint, our MEAD framework is quite damaging, creating a decrease in terms of AUROC↑%up to 6.5 percentage points and a maximal increase in FPR$_{\downarrow 95\%}$%of 5.3 percentage points. Under the single-armed setting, *FS* is extremely sensitive to the attacks generated by maximizing the Gini Impurity score.

FS	MEAD		ACE		KL		Gini		FR	
Norm L_1	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD1										
$\varepsilon = 5$	69.1	76.1	76.5	66.5	76.8	65.8	**68.2**	**77.9**	76.5	66.4
$\varepsilon = 10$	76.6	65.9	88.3	42.2	88.4	42.5	**74.8**	**68.4**	88.6	42.1
$\varepsilon = 15$	77.2	61.9	93.6	26.3	93.8	25.6	**76.7**	**63.8**	94.0	25.0
$\varepsilon = 20$	78.1	60.0	96.3	16.5	96.4	16.1	**76.3**	**63.2**	96.5	15.3
$\varepsilon = 25$	76.9	61.7	97.6	11.5	97.7	11.3	**74.7**	**64.9**	97.7	10.9
$\varepsilon = 30$	75.5	63.4	98.3	8.1	98.4	8.0	**72.6**	**66.8**	98.4	7.9
$\varepsilon = 40$	74.6	64.9	99.0	5.0	99.0	4.8	**71.4**	**68.1**	99.0	5.0
L_1 Average	75.4	64.8	92.8	25.1	92.9	24.9	**73.5**	**67.6**	92.9	24.6
Norm L_2	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD2										
$\varepsilon = 0.125$	**67.5**	**77.4**	74.6	68.9	75.1	67.6	68.9	76.9	74.8	69.2
$\varepsilon = 0.25$	75.9	66.1	87.3	45.0	87.4	45.0	**74.7**	**67.7**	87.4	45.4
$\varepsilon = 0.3125$	77.0	64.1	90.7	35.5	90.9	34.8	**76.1**	**65.7**	90.9	35.3
$\varepsilon = 0.5$	77.6	61.4	96.2	17.0	96.3	16.6	**75.7**	**64.6**	96.3	16.5
$\varepsilon = 1$	74.7	65.0	99.0	4.8	99.0	4.6	**71.2**	**67.9**	98.9	5.0
$\varepsilon = 1.5$	74.5	65.1	99.3	3.4	99.3	3.5	**71.1**	**68.0**	98.9	4.9
$\varepsilon = 2$	73.7	65.2	99.3	3.6	99.3	3.6	**71.1**	**68.0**	98.7	5.7
DeepFool										
No ε	**66.0**	**78.2**	**66.0**	**78.2**	**66.0**	**78.2**	**66.0**	**78.2**	**66.0**	**78.2**
CW2										
$\varepsilon = 0.01$	**86.7**	**46.3**	**86.7**	**46.3**	**86.7**	**46.3**	**86.7**	**46.3**	**86.7**	**46.3**
HOP										
$\varepsilon = 0.1$	**75.7**	**69.0**	**75.7**	**69.0**	**75.7**	**69.0**	**75.7**	**69.0**	**75.7**	**69.0**
L_2 Average	74.9	65.8	87.4	31.2	87.6	36.9	**73.7**	**67.2**	87.4	37.5
Norm L_∞	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGDi, FGSM, BIM										
$\varepsilon = 0.03125$	**59.6**	**79.0**	75.2	63.1	78.5	57.9	66.1	73.6	77.4	61.3
$\varepsilon = 0.0625$	**53.7**	**80.3**	71.9	63.8	76.3	59.5	60.2	77.1	77.2	57.6
$\varepsilon = 0.25$	**49.4**	**83.2**	72.8	60.0	78.2	53.9	55.2	81.6	76.8	53.8
$\varepsilon = 0.5$	**54.1**	**78.8**	82.7	39.9	86.4	36.5	57.2	78.2	78.6	52.3
PGDi, FGSM, BIM, SA										
$\varepsilon = 0.125$	**46.5**	**86.3**	64.3	71.6	69.7	68.1	51.5	85.2	70.8	66.3
PGDi, FGSM, BIM, CWi										
$\varepsilon = 0.3125$	**52.7**	**79.3**	71.1	62.2	75.7	58.2	58.9	77.2	73.4	59.9
L_∞ Average	**52.7**	**81.1**	73.0	60.1	77.5	55.7	58.2	78.8	75.7	58.5
No norm	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
STA										
No ε	**62.7**	**82.5**	**62.7**	**82.5**	**62.7**	**82.5**	**62.7**	**82.5**	**62.7**	**82.5**
No norm Average	**62.7**	**82.5**	**62.7**	**82.5**	**62.7**	**82.5**	**62.7**	**82.5**	**62.7**	**82.5**

Table B.5: Performances on *FS* per objective and in MEAD on CIFAR10. The worst results among all the settings is in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

MagNet

In Table B.6, we show the result of our MEAD framework on CIFAR10, evaluated on *MagNet*.

MagNet is an unsupervised detection method. In most cases, on CIFAR10, the results using MEAD are close to the worst results for the single-armed settings. In other words, it seems that if an example generated using the worst loss is detected (usually the Fisher-Rao objective), then the samples generated using all the others are detected. The decrease in AUROC↑% between the worst single-armed setting and MEAD is up to 1.8 percentage points.

MagNet	MEAD		ACE		KL		Gini		FR	
Norm L_1	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD1										
$\varepsilon = 5$	43.9	96.6	43.6	96.6	43.7	96.5	**43.3**	**97.1**	43.6	96.7
$\varepsilon = 10$	47.8	95.2	47.9	95.1	47.6	94.9	**46.7**	**95.6**	**46.7**	**95.6**
$\varepsilon = 15$	49.8	94.3	50.0	94.3	49.8	94.1	**49.1**	94.5	**49.1**	**94.7**
$\varepsilon = 20$	50.6	93.8	50.8	93.7	50.6	93.5	50.7	93.3	**49.8**	**94.1**
$\varepsilon = 25$	51.1	93.1	51.4	93.0	51.1	92.8	52.5	91.7	**50.6**	**93.3**
$\varepsilon = 30$	51.6	92.4	51.9	92.1	51.7	91.8	53.7	90.2	**51.2**	**92.5**
$\varepsilon = 40$	**52.7**	**90.6**	53.3	89.6	53.1	89.2	54.5	89.7	**52.7**	89.8
L_1 Average	49.6	93.7	49.8	93.5	49.7	93.3	50.1	93.2	**49.1**	**93.8**
Norm L_2	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD2										
$\varepsilon = 0.125$	43.1	96.9	**42.6**	96.9	43.2	96.8	43.8	96.7	43.2	**97.0**
$\varepsilon = 0.25$	47.3	95.4	47.3	95.4	47.1	95.1	**45.6**	**95.7**	46.4	**95.7**
$\varepsilon = 0.3125$	48.7	94.8	48.8	94.7	48.7	94.5	**47.4**	95.0	48.0	**95.1**
$\varepsilon = 0.5$	50.6	93.8	50.7	93.6	50.5	93.4	50.5	93.3	**49.9**	**94.2**
$\varepsilon = 1$	**53.0**	**90.1**	53.7	88.7	53.6	88.4	54.4	89.6	53.3	88.8
$\varepsilon = 1.5$	55.3	88.3	59.3	79.2	59.1	78.8	**54.5**	**89.5**	59.2	80.9
$\varepsilon = 2$	56.9	88.2	67.2	64.4	67.2	64.0	**54.5**	**89.5**	63.7	79.3
DeepFool										
No ε	**51.1**	**94.7**	**51.1**	**94.7**	**51.1**	**94.7**	**51.1**	**94.7**	**51.1**	**94.7**
CW2										
$\varepsilon = 0.01$	**50.5**	**94.7**	**50.5**	**94.7**	**50.5**	**94.7**	**50.5**	**94.7**	**50.5**	**94.7**
HOP										
$\varepsilon = 0.1$	**52.2**	**93.8**	**52.2**	**93.8**	**52.2**	**93.8**	**52.2**	**93.8**	**52.2**	**93.8**
L_2 Average	50.9	93.1	52.3	89.6	52.3	89.4	**50.5**	**93.3**	51.8	91.4
Norm L_∞	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGDi, FGSM, BIM										
$\varepsilon = 0.03125$	**58.6**	**82.0**	60.0	80.0	60.4	79.0	60.4	80.8	59.0	81.0
$\varepsilon = 0.0625$	**74.6**	**51.2**	76.8	48.0	79.4	46.2	77.1	48.2	78.8	47.0
$\varepsilon = 0.25$	**97.0**	**5.2**	98.2	3.6	98.7	3.4	97.7	4.1	98.7	3.4
$\varepsilon = 0.5$	**98.0**	**3.5**	99.0	2.2	99.2	2.0	98.6	2.6	99.2	2.1
PGDi, FGSM, BIM, SA										
$\varepsilon = 0.125$	**87.0**	**40.0**	88.8	39.3	90.9	39.3	88.9	37.3	91.4	39.3
PGDi, FGSM, BIM, CWi										
$\varepsilon = 0.3125$	52.6	**94.5**	**52.5**	**94.5**	**52.5**	**94.5**	52.6	**94.5**	52.6	**94.5**
L_∞ Average	**78.0**	**46.1**	79.2	44.6	80.2	44.1	79.2	44.6	80.0	44.6
No norm	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
STA										
No ε	**79.9**	**45.7**	**79.9**	**45.7**	**79.9**	**45.7**	**79.9**	**45.7**	**79.9**	**45.7**
No norm Average	**79.9**	**45.7**	**79.9**	**45.7**	**79.9**	**45.7**	**79.9**	**45.7**	**79.9**	**45.7**

Table B.6: Performances on *MagNet* per objective and in MEAD on CIFAR10. The worst results among all the settings are shown in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

B.1.2 Additional Results on MNIST

Success of attacks

In Table B.7, we show the average and total numbers of successful attack per settings (MEAD, Adversarial Cross-Entropy, KL divergence, Gini Impurity score and Fisher-Rao loss) on the MNIST dataset. We can observe the same behavior in MNIST as in CIFAR10. The most harmful attacks for the classifier are the ones generated according to the Adversarial Cross-Entropy. The attacks generated thanks to the three other objectives have a similar strength.

Avg. Num. of Successful Attack / Tot. Num. of Attack					
Norm L_1	MEAD	ACE	KL	Gini	FR
PGD1					
$\varepsilon = 5$	0.06 / 4	0.02 / 1	0.02 / 1	0.01 / 1	0.01 / 1
$\varepsilon = 10$	0.23 / 4	0.09 / 1	0.06 / 1	0.03 / 1	0.05 / 1
$\varepsilon = 15$	0.77 / 4	0.31 / 1	0.20 / 1	0.10 / 1	0.16 / 1
$\varepsilon = 20$	1.50 / 4	0.58 / 1	0.38 / 1	0.22 / 1	0.33 /1
$\varepsilon = 25$	2.03 / 4	0.73 / 1	0.48 / 1	0.33 / 1	0.48 / 1
$\varepsilon = 30$	2.35 / 4	0.80 / 1	0.54 / 1	0.42 / 1	0.59 / 1
$\varepsilon = 40$	2.67 / 4	0.85 / 1	0.58 / 1	0.54 / 1	0.70 / 1
Norm L_2					
PGD2					
$\varepsilon = 0.125$	0.04 / 4	0.01 / 1	0.01 / 1	0.01 / 1	0.01 / 1
$\varepsilon = 0.25$	0.04 / 4	0.01 / 1	0.01 / 1	0.01 / 1	0.01 / 1
$\varepsilon = 0.3125$	0.05 / 4	0.01 / 1	0.01 / 1	0.01 / 1	0.01 / 1
$\varepsilon = 0.5$	0.07 / 4	0.02 / 1	0.02 / 1	0.01 / 1	0.02 / 1
$\varepsilon = 1$	0.29 / 4	0.12 / 1	0.08 / 1	0.04 / 1	0.06 / 1
$\varepsilon = 1.5$	0.99 / 4	0.40 / 1	0.26 / 1	0.13 / 1	0.21 / 1
$\varepsilon = 2$	1.75 / 4	0.63 / 1	0.44 / 1	0.28 / 1	0.40 / 1
DeepFool					
No ε	0.97 / 1	0.97 / 1	0.97 / 1	0.97 / 1	0.97 / 1
CW2					
$\varepsilon = 0.01$	0.74 / 1	0.74 / 1	0.74 / 1	0.74 / 1	0.74 / 1
HOP					
$\varepsilon = 0.1$	0.99 / 1	0.99 / 1	0.99 / 1	0.99 / 1	0.99 / 1
Norm L_∞					
PGDi, FGSM, BIM					
$\varepsilon = 0.03125$	0.14 / 12	0.04 / 3	0.03 / 3	0.04 / 3	0.03 / 3
$\varepsilon = 0.0625$	0.38 / 12	0.13 / 3	0.08 / 3	0.09 / 3	0.07 / 3
$\varepsilon = 0.125$	2.07 / 12	0.79 / 3	0.46 / 3	0.45 / 3	0.37 / 3
$\varepsilon = 0.25$	7.41 / 12	2.62 / 3	1.50 / 3	1.70/ 3	1.60 / 3
$\varepsilon = 0.5$	8.85 / 12	2.90 / 3	1.76 / 3	2.20 / 3	1.99 / 3
PGDi, FGSM, BIM, CWi, SA					
$\varepsilon = 0.3125$	9.73 / 14	4.34 / 5	3.17 / 5	3.51 / 5	3.34 / 5
No norm					
STA					
No ε	0.85 / 1	0.85 / 1	0.85 / 1	0.85 / 1	0.85 / 1

Table B.7: Average number of successful attacks per natural sample considered in the single-armed setting and MEAD (MNIST). The results are reported in the table together with the total number of attacks performed per natural sample (*Avg. / Tot.*). No norm denotes the group of attacks that do not depend on the norm constraint.

NSS

In Table B.8, we show the result of our MEAD framework on MNIST, evaluated on *NSS*. *NSS* is effective on MNIST, in particular when considering L_∞ threats. However, in this case, when $\varepsilon = 0.3125$, the performances decrease. This is not surprising as CWi and SA have different attack schemes from the ones in PGD/FGSM/BIM. *NSS* also loses some of its effectiveness with L_1 and L_2

NSS	MEAD		ACE		KL		Gini		FR	
Norm L_1	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%
PGD1										
$\varepsilon = 5$	**91.4**	**30.5**	92.1	24.6	92.3	27.2	93.0	24.2	92.0	26.9
$\varepsilon = 10$	**96.4**	**11.7**	96.6	10.8	96.7	10.7	97.3	8.0	97.1	8.5
$\varepsilon = 15$	**97.3**	**8.6**	97.5	8.0	97.5	8.0	98.0	4.2	97.5	7.7
$\varepsilon = 20$	**97.9**	**5.3**	98.0	4.7	98.0	4.6	98.2	3.3	98.1	3.9
$\varepsilon = 25$	**98.2**	**3.2**	98.3	**3.2**	98.3	**3.2**	98.3	3.1	98.3	**3.2**
$\varepsilon = 30$	**98.3**	**3.1**	**98.3**	**3.1**	**98.3**	**3.1**	**98.3**	**3.1**	**98.3**	**3.1**
$\varepsilon = 40$	**98.4**	**3.1**	**98.4**	**3.1**	**98.4**	**3.1**	**98.4**	**3.1**	**98.4**	**3.1**
L_1 Average	**96.8**	**9.4**	97.0	8.2	97.1	8.6	97.4	7.0	97.1	8.1
Norm L_2	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%
PGD2										
$\varepsilon = 0.125$	**80.4**	55.3	81.3	55.2	81.3	49.7	82.3	51.4	81.2	**56.2**
$\varepsilon = 0.25$	**86.4**	**42.0**	87.6	38.4	87.9	39.7	89.0	40.3	86.9	41.8
$\varepsilon = 0.3125$	**88.7**	33.8	89.8	33.1	90.1	32.5	90.9	32.7	89.1	**37.6**
$\varepsilon = 0.5$	**92.6**	21.7	92.9	20.6	93.1	**22.2**	94.7	17.5	93.2	20.9
$\varepsilon = 1$	**96.8**	**10.0**	96.9	9.2	97.0	9.0	97.5	7.0	97.2	8.1
$\varepsilon = 1.5$	**97.5**	**8.1**	97.6	7.5	97.6	7.1	98.0	4.5	97.6	7.3
$\varepsilon = 2$	**98.0**	**4.6**	98.1	4.1	98.1	3.5	98.1	3.7	98.1	3.3
DeepFool										
No ε	**97.8**	**4.8**	**97.8**	**4.8**	**97.8**	**4.8**	**97.8**	**4.8**	**97.8**	**4.8**
CW2										
$\varepsilon = 0.01$	**66.9**	**81.9**	**66.9**	**81.9**	**66.9**	**81.9**	**66.9**	**81.9**	**66.9**	**81.9**
HOP										
$\varepsilon = 0.1$	**98.3**	**3.1**	**98.3**	**3.1**	**98.3**	**3.1**	**98.3**	**3.1**	**98.3**	**3.1**
L_2 Average	**90.3**	**26.5**	90.7	25.8	90.8	25.4	91.4	23.7	90.6	**26.5**
Norm L_∞	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%
PGDi, FGSM, BIM										
$\varepsilon = 0.03125$	**93.4**	**9.6**	94.5	9.5	94.7	9.5	95.3	9.5	94.5	9.5
$\varepsilon = 0.0625$	**92.5**	**9.6**	93.2	**9.6**	93.6	**9.6**	93.5	**9.6**	93.3	**9.6**
$\varepsilon = 0.25$	**92.2**	**9.6**	93.1	**9.6**	93.2	**9.6**	93.7	**9.6**	93.5	**9.6**
$\varepsilon = 0.5$	**91.5**	**9.6**	92.5	**9.6**	92.3	**9.6**	93.8	**9.6**	93.0	**9.6**
PGDi, FGSM, BIM, CWi, SA										
$\varepsilon = 0.3125$	73.9	79.0	74.2	79.1	**73.5**	**79.6**	73.7	79.4	74.3	79.2
L_∞ Average	**88.7**	23.5	89.5	23.5	89.5	**23.6**	90.0	**23.6**	89.8	23.5
No norm	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%	AUROC↑%%	$FPR_{95\%}$↓%%
STA										
No ε	**87.1**	**57.8**	**87.1**	**57.8**	**87.1**	**57.8**	**87.1**	**57.8**	**87.1**	**57.8**
No norm Average	**87.1**	**57.8**	**87.1**	**57.8**	**87.1**	**57.8**	**87.1**	**57.8**	**87.1**	**57.8**

Table B.8: Performances on *NSS* per objective and in MEAD on MNIST. The worst results among all the settings is in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

threats. Note that all the single-armed settings behave quite similarly: this is probably due to the computation of the *Natural Scene Statistics*, which are not meaningful for perturbed images. Therefore, it is not surprising that the decrease in AUROC↑% considering MEAD is less than one percentage point compared to the worst single-armed setting.

KD-BU

In Table B.9, we show the result of our MEAD framework on MNIST, evaluated on *KD-BU*.

KD-BU	MEAD		ACE		KL		Gini		FR	
Norm L_1	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD1										
$\varepsilon = 5$	**45.2**	**95.7**	60.9	92.5	56.6	93.8	61.5	92.4	52.8	94.6
$\varepsilon = 10$	**46.4**	**95.6**	58.3	93.3	59.9	92.9	59.2	93.2	59.4	93.0
$\varepsilon = 15$	**45.5**	**95.7**	58.3	93.4	59.3	93.0	59.7	93.0	58.2	93.4
$\varepsilon = 20$	**45.5**	**95.7**	59.7	93.1	58.7	93.3	61.3	92.7	59.6	93.1
$\varepsilon = 25$	**45.8**	**95.7**	60.3	93.0	60.2	92.9	61.7	92.7	60.5	92.9
$\varepsilon = 30$	**45.7**	**95.7**	60.8	92.9	60.4	92.9	62.9	92.3	60.3	93.0
$\varepsilon = 40$	**44.8**	**95.8**	61.2	92.8	60.2	93.0	63.5	92.3	61.2	92.8
L_1 Average	**45.6**	**95.7**	59.9	93.0	59.3	93.1	61.4	92.7	58.9	93.3
Norm L_2	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD2										
$\varepsilon = 0.125$	**43.0**	**96.0**	59.8	92.8	58.1	93.4	55.8	94.0	56.0	94.0
$\varepsilon = 0.25$	**44.1**	**95.9**	60.4	92.6	57.7	93.5	56.0	94.0	60.6	92.6
$\varepsilon = 0.3125$	**45.3**	**95.7**	55.6	94.0	59.2	93.0	55.7	94.0	59.3	93.0
$\varepsilon = 0.5$	**43.9**	**95.9**	57.0	93.7	55.8	94.0	58.3	93.4	55.5	94.0
$\varepsilon = 1$	**46.6**	**95.6**	59.2	93.1	58.1	93.4	59.4	93.0	58.4	93.3
$\varepsilon = 1.5$	**45.8**	**95.7**	58.8	93.3	60.0	92.9	58.9	93.3	58.8	93.2
$\varepsilon = 2$	**46.7**	**95.6**	60.0	93.0	59.5	93.1	61.0	92.8	60.9	92.7
DeepFool										
No ε	**62.9**	**92.4**	**62.9**	**92.4**	**62.9**	**92.4**	**62.9**	**92.4**	**62.9**	**92.4**
CW2										
$\varepsilon = 0.01$	**62.5**	**92.5**	**62.5**	**92.5**	**62.5**	**92.5**	**62.5**	**92.5**	**62.5**	**92.5**
HOP										
$\varepsilon = 0.1$	**62.6**	**92.5**	**62.6**	**92.5**	**62.6**	**92.5**	**62.6**	**92.5**	**62.6**	**92.5**
L_2 Average	**50.3**	**94.8**	59.9	93.0	59.7	93.1	59.3	93.2	59.8	93.0
Norm L_∞	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGDi, FGSM, BIM										
$\varepsilon = 0.03125$	**34.5**	**96.7**	42.0	96.1	44.0	95.9	48.1	95.4	42.9	96.0
$\varepsilon = 0.0625$	**33.6**	**96.8**	41.0	96.2	44.2	95.9	47.6	95.5	44.1	95.9
$\varepsilon = 0.25$	**34.2**	**96.7**	44.6	95.8	44.5	95.8	52.2	94.8	45.9	95.6
$\varepsilon = 0.5$	**34.0**	**96.6**	44.5	95.7	44.8	95.7	51.2	94.9	46.0	95.6
PGDi, FGSM, BIM, CWi, SA										
$\varepsilon = 0.3125$	**34.2**	**96.7**	41.7	96.1	45.8	95.7	44.0	95.8	45.6	95.7
L_∞ Average	**34.1**	**96.7**	42.8	96.0	44.7	95.8	48.6	95.3	44.9	95.8
No norm	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
STA										
No ε	**76.0**	**88.2**	**76.0**	**88.2**	**76.0**	**88.2**	**76.0**	**88.2**	**76.0**	**88.2**
No norm Average	**76.0**	**88.2**	**76.0**	**88.2**	**76.0**	**88.2**	**76.0**	**88.2**	**76.0**	**88.2**

Table B.9: Performances on *KD-BU* per objective and in MEAD on MNIST. The worst results among all the settings are shown in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

The *KD-BU* method is the least effective one at detecting the adversarial samples under the MEAD framework. The decrease of AUROC↑% can go up to 23 percentage points. Fisher-Rao and Adversarial Cross-Entropy-based attacks seem to be the toughest to detect for *KD-BU* detectors.

LID

In Table B.10, we show the result of our MEAD framework on MNIST, evaluated on *LID*.

LID	MEAD		ACE		KL		Gini		FR	
Norm L_1	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD1										
$\varepsilon = 5$	88.1	41.5	89.6	36.1	88.6	41.7	88.4	39.3	**87.5**	**46.9**
$\varepsilon = 10$	**83.2**	**48.8**	86.8	41.9	87.1	41.3	89.4	35.7	86.9	42.1
$\varepsilon = 15$	**83.1**	48.8	84.3	41.9	84.2	46.5	87.1	42.3	83.7	**49.8**
$\varepsilon = 20$	**77.7**	**57.7**	83.8	47.2	84.6	46.6	90.1	35.5	84.5	47.8
$\varepsilon = 25$	**78.5**	**58.7**	83.0	51.9	83.5	56.8	91.4	32.4	83.6	51.2
$\varepsilon = 30$	**74.3**	**65.6**	80.8	57.4	81.8	56.8	92.7	29.1	82.7	54.9
$\varepsilon = 40$	**74.5**	**63.5**	77.5	60.8	78.4	60.1	93.8	26.5	80.1	58.9
L_1 Average	**79.9**	**54.9**	83.7	48.2	84.0	50.0	90.4	52.1	84.1	50.2
Norm L_2	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD2										
$\varepsilon = 0.125$	87.7	47.7	88.0	47.5	87.6	47.4	**86.8**	46.7	**86.8**	**49.1**
$\varepsilon = 0.25$	88.0	40.5	89.1	45.2	**87.9**	**49.2**	88.1	45.2	88.0	44.6
$\varepsilon = 0.3125$	88.4	39.7	89.6	44.0	87.9	46.0	**87.6**	**53.1**	88.1	45.4
$\varepsilon = 0.5$	**88.0**	38.1	90.0	33.8	88.9	35.6	88.1	**44.5**	88.1	41.1
$\varepsilon = 1$	**80.1**	**55.3**	86.8	42.3	87.0	41.8	88.0	39.1	86.7	43.2
$\varepsilon = 1.5$	**81.9**	**51.3**	84.8	46.0	84.5	47.2	87.4	42.0	84.0	48.7
$\varepsilon = 2$	**81.1**	**53.6**	85.2	46.4	84.8	47.2	89.2	38.1	85.6	46.2
DeepFool										
No ε	**87.9**	**42.1**	**87.9**	**42.1**	**87.9**	**42.1**	**87.9**	**42.1**	**87.9**	**42.1**
CW2										
$\varepsilon = 0.01$	**83.6**	**52.6**	**83.6**	**52.6**	**83.6**	**52.6**	**83.6**	**52.6**	**83.6**	**52.6**
HOP										
$\varepsilon = 0.1$	**89.3**	**41.0**	**89.3**	**41.0**	**89.3**	**41.0**	**89.3**	**41.0**	**89.3**	**41.0**
L_2 Average	**85.6**	**46.2**	87.4	44.1	87.0	45.1	87.6	44.4	86.1	45.4
Norm L_∞	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGDi, FGSM, BIM										
$\varepsilon = 0.03125$	87.5	44.1	89.8	34.5	87.6	45.3	89.5	40.0	**86.9**	**46.7**
$\varepsilon = 0.0625$	**84.7**	**45.5**	88.3	37.6	88.1	36.7	88.3	38.0	87.7	42.4
$\varepsilon = 0.125$	**80.6**	**52.1**	85.3	43.1	85.6	43.9	84.2	45.3	84.4	46.7
$\varepsilon = 0.25$	**74.1**	**63.0**	83.7	48.9	83.7	49.5	91.5	32.9	85.6	47.6
$\varepsilon = 0.5$	**65.4**	**66.8**	74.4	58.8	75.2	58.8	92.3	32.9	72.3	61.4
PGDi, FGSM, BIM, CWi, SA										
$\varepsilon = 0.3125$	**74.8**	**59.1**	78.0	55.1	81.3	52.5	86.2	43.9	80.9	52.4
L_∞ Average	**77.9**	**55.1**	83.3	46.3	83.6	47.8	88.7	38.8	83.0	49.5
No norm	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
STA										
No ε	**98.1**	**8.2**	**98.1**	**8.2**	**98.1**	**8.2**	**98.1**	**8.2**	**98.1**	**8.2**
No norm Average	**98.1**	**8.2**	**98.1**	**8.2**	**98.1**	**8.2**	**98.1**	**8.2**	**98.1**	**8.2**

Table B.10: Performances on *LID* per objective and in MEAD on MNIST. The worst results among all the settings are shown in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

LID is quite effective in detecting STA. Contrary to the other methods, *LID* has more difficulty detecting attacks with significant perturbations. The maximum decrease in AUROC↑% considering MEAD is slightly higher than 8 percentage points. Even if the results seem to be quite similar among the single-armed, the *LID*-based detector trained on MNIST seems more vulnerable to the attacks generated thanks to the Kullback-Leibler divergence on L_1-norm-based attacks and sensitive to the Fisher-Rao distance under L_2 threats.

FS

In Table B.11, we show the result of our MEAD framework on MNIST, evaluated on *FS*.

FS	MEAD		ACE		KL		Gini		FR	
Norm L_1	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD1										
$\varepsilon = 5$	**59.2**	88.1	63.3	85.4	62.6	87.1	**59.2**	84.7	59.8	**89.5**
$\varepsilon = 10$	**72.0**	**76.3**	76.8	68.6	76.9	67.8	73.0	70.0	75.6	69.8
$\varepsilon = 15$	86.5	**56.4**	90.7	45.2	90.9	44.3	**84.9**	54.2	90.7	44.5
$\varepsilon = 20$	**87.9**	**54.1**	92.3	41.3	92.5	40.0	90.8	42.3	92.5	40.5
$\varepsilon = 25$	**86.6**	**56.0**	90.4	46.1	90.5	45.9	92.0	39.0	90.9	46.5
$\varepsilon = 30$	**83.9**	**60.4**	88.0	53.0	88.3	50.9	92.0	39.3	89.5	49.9
$\varepsilon = 40$	**72.3**	**76.0**	82.5	63.8	82.7	63.8	90.8	41.8	84.6	60.8
L_1 Average	**79.8**	**66.8**	83.4	57.6	83.5	57.1	83.2	53.0	83.4	57.4
Norm L_2	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD2										
$\varepsilon = 0.125$	**56.6**	90.5	58.3	89.2	57.7	**91.4**	57.0	89.4	57.4	87.9
$\varepsilon = 0.25$	56.2	**90.3**	58.3	85.4	58.5	85.8	58.0	89.3	**55.9**	89.1
$\varepsilon = 0.3125$	**57.1**	88.1	60.3	85.6	60.1	86.6	60.8	86.2	57.7	**91.3**
$\varepsilon = 0.5$	**59.5**	85.4	62.9	82.7	62.4	85.4	64.0	81.7	60.5	**85.9**
$\varepsilon = 1$	76.9	**67.1**	80.2	62.8	80.2	62.5	**76.4**	64.5	79.6	61.0
$\varepsilon = 1.5$	89.2	47.0	92.9	33.3	93.1	33.2	**86.9**	**47.2**	92.8	35.5
$\varepsilon = 2$	**88.8**	**50.7**	92.4	40.5	92.7	39.1	91.2	41.7	93.2	36.3
DeepFool										
No ε	**88.3**	**52.0**	**88.3**	**52.0**	**88.3**	**52.0**	**88.3**	**52.0**	**88.3**	**52.0**
CW2										
$\varepsilon = 0.01$	**68.6**	**81.7**	**68.6**	**81.7**	**68.6**	**81.7**	**68.6**	**81.7**	**68.6**	**81.7**
HOP										
$\varepsilon = 0.1$	**93.4**	**36.6**	**93.4**	**36.6**	**93.4**	**36.6**	**93.4**	**36.6**	**93.4**	**36.6**
L_2 Average	**73.5**	**69.0**	75.6	65.0	75.5	65.4	74.5	67.0	74.7	65.7
Norm L_∞	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGDi, FGSM, BIM										
$\varepsilon = 0.03125$	**55.0**	**90.4**	59.6	87.8	58.1	88.7	58.9	85.6	57.0	88.4
$\varepsilon = 0.0625$	**62.8**	**83.5**	68.4	76.0	67.4	75.9	64.3	80.5	67.2	76.0
$\varepsilon = 0.25$	**96.7**	**17.9**	98.6	6.7	98.7	5.7	**96.7**	17.4	99.2	3.4
$\varepsilon = 0.5$	**82.2**	**60.0**	91.9	37.7	91.9	37.1	90.2	43.9	93.0	34.4
PGDi, FGSM, BIM, CWi, SA										
$\varepsilon = 0.3125$	85.1	65.8	85.7	64.8	85.0	65.7	**84.9**	**66.1**	85.7	64.9
L_∞ Average	**76.4**	**63.5**	80.8	54.6	80.2	54.6	79.0	58.7	80.4	58.2
No norm	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
STA										
No ε	**61.5**	**85.9**	**61.5**	**85.9**	**61.5**	**85.9**	**61.5**	**85.9**	**61.5**	**85.9**
No norm Average	**61.5**	**85.9**	**61.5**	**85.9**	**61.5**	**85.9**	**61.5**	**85.9**	**61.5**	**85.9**

Table B.11: Performances on *FS* per objective and in MEAD on MNIST. The worst results among all the settings is in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

Despite having trouble detecting attacks with a small maximal perturbation ε, *FS* detectors are not that bad at detecting adversarial examples. The attacks based on the Gini Impurity Score are the least detected ones among all the single-armed settings. The decrease in terms of AUROC↑% is, in that case, 8 percentage points at most.

MagNet

In Table B.12, we show the result of our MEAD framework on MNIST, evaluated on *MagNet*.

MagNet	MEAD		ACE		KL		Gini		FR	
Norm L_1	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD1										
$\varepsilon = 5$	**87.6**	**37.1**	88.5	35.1	88.8	36.8	88.7	34.6	87.7	36.6
$\varepsilon = 10$	**99.2**	**2.7**	99.3	2.3	**99.2**	2.5	99.4	1.9	**99.2**	2.3
$\varepsilon = 15$	**99.9**	**0.2**	**99.9**	0.1	**99.9**	**0.2**	100.0	0.1	**99.9**	0.1
$\varepsilon = 20$	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**
$\varepsilon = 25$	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**
$\varepsilon = 30$	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**
$\varepsilon = 40$	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**
L_1 Average	**98.1**	5.7	98.2	5.4	98.3	**5.6**	98.3	5.2	**98.1**	**5.6**
Norm L_2	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGD2										
$\varepsilon = 0.125$	64.3	82.4	65.6	80.4	65.5	82.5	65.7	80.1	**63.0**	**84.0**
$\varepsilon = 0.25$	74.4	**68.3**	75.5	66.8	76.6	66.4	77.2	68.0	**73.4**	67.0
$\varepsilon = 0.3125$	79.7	61.1	81.0	56.7	82.0	56.5	81.9	**61.3**	**78.4**	57.1
$\varepsilon = 0.5$	**90.8**	**32.2**	91.9	30.8	91.9	31.0	91.7	**32.2**	91.0	31.1
$\varepsilon = 1$	99.1	3.1	99.6	1.6	99.5	1.7	**98.2**	**8.4**	99.5	1.8
$\varepsilon = 1.5$	99.8	0.3	100.0	0.1	100.0	0.1	**99.5**	**1.6**	100.0	0.1
$\varepsilon = 2$	99.9	0.1	100.0	0.0	100.0	0.0	**99.7**	**0.4**	100.0	0.0
DeepFool										
No ε	**99.4**	**1.1**	**99.4**	**1.1**	**99.4**	**1.1**	**99.4**	**1.1**	**99.4**	**1.1**
CW2										
$\varepsilon = 0.01$	**92.8**	**38.3**	**92.8**	**38.3**	**92.8**	**38.3**	**92.8**	**38.3**	**92.8**	**38.3**
HOP										
$\varepsilon = 0.1$	**99.9**	**0.0**	**99.9**	**0.0**	**99.9**	**0.0**	**99.9**	**0.0**	**99.9**	**0.0**
L_2 Average	90.0	28.7	90.6	27.6	90.8	27.8	90.6	**29.1**	**89.7**	28.1
Norm L_∞	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
PGDi, FGSM, BIM										
$\varepsilon = 0.03125$	**100.0**	**0.1**	**100.0**	0.0	**100.0**	0.0	**100.0**	**0.1**	**100.0**	0.0
$\varepsilon = 0.0625$	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**
$\varepsilon = 0.25$	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**
$\varepsilon = 0.5$	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**	**100.0**	**0.0**
PGDi, FGSM, BIM, CWi, SA										
$\varepsilon = 0.3125$	92.6	51.4	92.6	51.4	**92.2**	**53.1**	92.3	52.5	92.5	51.8
L_∞ Average	98.5	10.3	98.5	10.3	**98.4**	**10.6**	98.5	10.5	98.5	10.4
No norm	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%	AUROC↑%%	FPR$_{95\%}$↓%%
STA										
No ε	**86.9**	**74.3**	**86.9**	**74.3**	**86.9**	**74.3**	**86.9**	**74.3**	**86.9**	**74.3**
No norm Average	**86.9**	**74.3**	**86.9**	**74.3**	**86.9**	**74.3**	**86.9**	**74.3**	**86.9**	**74.3**

Table B.12: Performances on *MagNet* per objective and in MEAD on MNIST. The worst results among all the settings are shown in **bold**; the ones in the single-armed setting is underlined. No norm denotes the group of attacks that do not depend on the norm constraint.

MagNet is effective on MNIST. It is pretty close to the perfect detector for L_∞ and L_1 attacks. Anyway, the Fisher-Rao-based attacks are the most disruptive ones for such detectors. Similar to the CIFAR10 case, the results using MEAD are quite close to the worst single-armed setting case. The decrease in AUROC↑% between MEAD and the worst single-armed setting is at most 0.5 percentage points.

APPENDIX C

Appendix to Chapter 5

C.1 On the optimization of Eq. (5.5)

The maximization problem in Eq. (5.5) is well-posed given that the mutual information is a concave function of $\omega \in \Omega$. Although from the theoretical point of view, Eq. (5.5) guarantees the optimal solution for the average regret minimization problem, in practice, we have to deal with some technical limitations. For the optimization of Eq. (5.5), we rely on the `SciPy` [VGO+20] library, package `optimize`, function `minimize`[1] which uses the *Sequential Least Squares Programming* (SLSQP) algorithm to find the optimum. This algorithm relies on local optimization and is particularly straightforward when dealing with non-linear equations and equality and inequality constraints, as in our case. Overall, we obtained the satisfactory results provided in the paper by assigning default values to all the parameters and by setting a uniform distribution $[\omega_1, \omega_2, \omega_3, \omega_4] = [.25, .25, .25, .25]$ as the initial point in the solutions space.

Although these results are satisfactory and confirm the value of the sound theoretical framework, we propose in Section 5.1. We are well aware that, in some cases, as in Fig. 5.2a, the proposed aggregation slightly underperforms in terms of accuracy w.r.t. the best detector in the set of allowed detectors. In this regard, we would like to raise a couple of points that are interesting for practitioners and possible future research:

1. For each input sample, we solve one different optimization problem: although the algorithm above always reaches the end with a success state,

[1] Therefore we invert the sign of the objective function.

given the finite amount of iterations and the tolerance which decides the stopping criterion, further sample-by-sample parameter optimization may be required. At this time, we have not delved into the problem, and we leave this for future research.

2. The hard decisions made by the single detectors only depend on the argmax of their soft-probabilities. On the contrary, the optimization in Eq. (5.5) considers the complete soft-probability distributions output by every single detector. Indeed, although the hard decision on two randomly considered samples can be right for both, often, the confidence in these decisions can be very different (i.e., two correctly classified samples may have utterly different associated soft probabilities). Further research on how differently accurate detectors influence the optimization in Eq. (5.5) is left for future work.

C.2 Supplementary Results of Section 5.2

In the following, we provide further discussions on the experiments in Section 5.2 that have not been included in the main chapter.

Experimental environment

We run each experiment on a machine equipped with an Intel(R) Xeon(R) Gold 6226 CPU, 2.70GHz clock frequency, and a Tesla V100-SXM2-32GB GPU.

Time measurements

Training 1 single detector in our method	`1h45m10s`
Evaluating the optimization in our method	`1m35s` (for one attack)
Training NSS	`3m30s`
Evaluating NSS	`20s` (for one attack)
On the largest set of simultaneous attacks (13 attacks):	
Ours	`1m35s * 13 ~ 21m`
NSS	`20s * 13 ~ 4m`

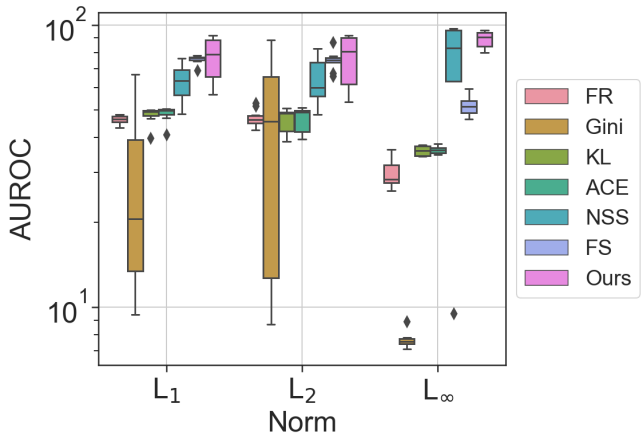


Figure C.1: The *shallow* detectors are named after the loss function used to craft the attacks they are trained to detect. Overall, the SOTA methods NSS and FS outperform all the individual shallow detectors. The aggregation we propose allows using the shallow models to attain a detector whose performance is consistently comparable and, in many cases, better than SOTA.

C.2.1 On the MEAD framework

State-of-the-art detectors

Chapter 4 suggests NSS [KFHD20] and FS [XEQ18] as the most robust methods in the simultaneous attacks detection scheme (i.e., MEAD). We remind that NSS is a supervised method that extracts the *natural scene statistics* of the natural and adversarial examples to train a SVM. On the contrary, FS is an unsupervised method that uses *feature squeezing* (i.e., reducing the color depth of images and using smoothing to reduce the variation among the pixels) to compare the model's predictions.

In particular, we choose NSS as a method to compare for multiple reasons:

1. NSS achieves the best overall score in terms of AUROC↑% and FPR$\downarrow_{95\%}$% among the SOTA against simultaneous attacks (cf. Table 4.3).

2. NSS achieves the best score in terms of AUROC↑% and FPR$\downarrow_{95\%}$% under the L_∞ norm where the biggest group of simultaneous attacks are evaluated (see Table 5.1). This is stressed in the plots in Fig. C.1. Moreover, FS reaches better performance w.r.t. the proposed method only with PGD1 and PGD2 when the perturbation magnitude is small and in CW2.

Table C.1: Comparison between Ours and Ours+FS on CIFAR10. The * symbol means the perturbation mechanism is executed in parallel four times starting from the same original clean sample, each time using one of the objective losses between ACE Eq. (4.2), KL Eq. (4.3), FR Eq. (4.4), Gini Eq. (4.5). We focus only on the cases in which the proposed method is outperformed by the corresponding competitors.

	CIFAR10			
	Ours		Ours+FS	
	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%
Norm L_1				
PGD1*				
$\varepsilon = 5$	62.1	87.1	**69.4**	**74.5**
$\varepsilon = 10$	56.8	90.6	**76.8**	**64.5**
$\varepsilon = 15$	69.3	84.4	**77.6**	**60.3**
Norm L_2				
PGD2*				
$\varepsilon = 0.125$	63.9	85.4	**67.9**	**76.4**
$\varepsilon = 0.25$	57.1	90.5	**76.0**	**64.7**
$\varepsilon = 0.3125$	61.0	88.9	**77.2**	**62.9**
CW2				
$\varepsilon = 0.01$	53.4	92.2	**86.4**	**46.8**

3. The case study for our aggregator in the experimental section is based on supervised detectors as a consequence the comparison with a supervised detector was a natural choice.

For the sake of completeness, the performances of NSS and FS under MEAD are given in Fig. C.1. As shown before for Ours+NSS, in Table C.1 we propose an analysis of the performance of our method before and after adding the FS unsupervised detection mechanism to the pull of available detectors, showing a stark improvement in the latter case.

Attacks

We want to emphasize that, differently from the literature, we are the first to consider a defense mechanism against the simultaneous attack setting in which we detect attacks based on four different losses. More specifically, for each 'clean dataset' (in our case CIFAR10 and SVHN):

- No. of adversarial examples generated with:

Table C.2: Simultaneous attacks detection: NSS on CIFAR10. We train NSS on natural and adversarial examples created with PGD algorithm and L_∞ norm constraint. The perturbation magnitude ε is shown in the columns. We indicate in **bold** the best result.

	NSS											
	0.03125		0.0625		0.125		0.25		0.3125		0.5	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1												
PGD1												
$\varepsilon = 5$	**54.8**	**94.2**	47.7	94.7	46.6	95.6	46.8	95.5	47.0	95.4	46.5	95.6
$\varepsilon = 10$	**54.0**	**90.3**	53.4	90.8	51.6	94.3	50.4	94.9	50.4	94.9	50.9	94.7
$\varepsilon = 15$	**58.8**	**86.8**	58.1	87.4	55.8	92.8	53.8	94.2	53.2	94.4	54.5	93.7
$\varepsilon = 20$	**63.5**	**82.3**	62.7	82.7	60.1	90.7	57.4	93.2	56.7	93.6	58.2	92.3
$\varepsilon = 25$	**67.7**	**77.2**	66.8	78.4	64.0	87.8	61.0	92.0	60.1	92.6	61.9	90.6
$\varepsilon = 30$	**71.4**	**73.4**	70.5	73.5	67.6	83.7	64.4	90.4	63.4	91.4	65.4	88.2
$\varepsilon = 40$	**76.1**	**67.3**	75.3	68.0	72.6	75.4	69.4	87.2	68.5	88.9	70.4	83.4
Norm L_2												
PGD2												
$\varepsilon = 0.125$	**48.3**	**94.3**	47.5	94.8	46.6	95.6	46.7	95.5	47.1	95.4	46.5	95.6
$\varepsilon = 0.25$	**53.2**	**91.2**	52.6	91.6	50.9	94.6	50.0	95.0	50.0	95.0	50.3	94.8
$\varepsilon = 0.3125$	**55.8**	**89.2**	55.2	89.9	53.3	93.7	51.7	94.6	51.5	94.7	52.3	94.3
$\varepsilon = 0.5$	**63.3**	**82.6**	62.6	83.0	60.0	90.7	57.4	93.2	56.7	93.5	58.2	92.4
$\varepsilon = 1$	**76.4**	**67.5**	75.7	67.8	73.1	75.0	70.1	86.7	69.2	88.5	71.0	83.0
$\varepsilon = 1.5$	**81.0**	63.0	80.5	**62.7**	78.5	63.5	76.2	80.7	75.6	83.2	76.9	74.4
$\varepsilon = 2$	**82.6**	62.3	82.1	**61.6**	80.6	62.5	78.6	78.5	78.1	81.2	79.1	72.1
DeepFool												
No ε	**57.0**	**91.7**	56.7	**91.7**	55.6	93.6	54.6	94.1	54.2	94.3	54.7	94.0
CW2												
$\varepsilon = 0.01$	**56.4**	**90.8**	55.9	90.9	54.5	93.7	53.4	94.3	53.0	94.5	53.6	94.1
HOP												
$\varepsilon = 0.1$	**66.1**	**87.0**	65.1	88.2	63.0	91.3	61.2	92.6	60.8	92.9	61.6	92.1
Norm L_∞												
PGDi, FGSM, BIM												
$\varepsilon = 0.03125$	**83.0**	55.3	82.1	**55.2**	80.3	57.8	77.4	77.0	76.8	81.3	78.3	65.4
$\varepsilon = 0.0625$	**96.0**	**17.2**	94.6	17.4	94.9	19.2	94.3	21.6	94.4	21.1	94.4	21.1
$\varepsilon = 0.25$	**97.3**	**0.6**	94.7	5.9	96.5	2.5	96.9	1.7	97.2	1.1	96.7	2.1
$\varepsilon = 0.5$	**82.5**	**100.0**	80.4	**100.0**	81.9	**100.0**	82.2	**100.0**	82.4	**100.0**	82.0	**100.0**
PGDi, FGSM, BIM, SA												
$\varepsilon = 0.125$	9.4	**99.9**	10.4	100.0	26.2	**99.9**	30.9	100.0	**33.8**	100.0	27.3	100.0
PGDi, FGSM, BIM, CWi												
$\varepsilon = 0.3125$	**63.2**	99.1	62.7	**99.0**	61.9	99.3	60.9	99.5	60.5	99.5	61.2	99.4
No norm												
STA												
No ε	88.5	38.8	92.0	25.1	92.1	22.4	**93.3**	**18.3**	92.7	19.6	92.7	19.7

- L_1 norm: 7 (no. of ε) * 1 (PGD algorithm) * 4 (no. of losses) = 28 ('adversarial datasets')
- L_2 norm: 7 (no. of ε) * 1 (PGD algorithm) * 4 (no. of losses) + 3 (CW2, HOP, DeepFool) = 31 ('adversarial datasets')
- L_∞ norm: 6 (no. of ε) * 3 (PGD, FGSM, BIM algorithms) * 4 (no. of losses) + 2 = 74 ('adversarial datasets')
- No norm: 1 ('adversarial dataset')

=> For a total of `28 + 31 + 74 + 1 = 134` 'adversarial datasets' for each 'clean dataset'.

Moreover, it is interesting to notice that the experiments on CIFAR10 and SVHN represent a satisfying choice to show that state-of-the-art detection mechanisms struggle to maintain good performance when faced with the framework of simultaneous attacks. That said, we leave the evaluation of larger datasets as future

Table C.3: Simultaneous attacks detection: the proposed method on CIFAR10. We train NSS on natural and adversarial examples created with PGD algorithm and L_∞ norm constraint. The perturbation magnitude ε is shown in the columns. We indicate in **bold** the best result.

	Ours											
	0.03125		0.0625		0.125		0.25		0.3125		0.5	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1												
PGD1												
$\varepsilon = 5$	**69.7**	82.5	65.5	**81.5**	62.1	87.1	56.3	93.8	53.2	94.8	48.5	95.5
$\varepsilon = 10$	62.3	**83.3**	**62.7**	86.3	56.8	90.6	52.1	94.7	52.9	94.6	50.9	95.0
$\varepsilon = 15$	66.6	**72.7**	**73.9**	77.9	69.3	84.4	65.5	89.0	64.3	91.0	60.4	93.1
$\varepsilon = 20$	72.8	**58.0**	**83.7**	59.3	78.7	73.1	73.8	82.5	73.5	85.4	69.2	90.3
$\varepsilon = 25$	76.8	42.4	**89.4**	**35.9**	87.1	50.8	81.3	68.6	79.3	78.0	74.8	87.2
$\varepsilon = 30$	79.1	31.1	**91.7**	**21.4**	90.3	35.4	84.3	61.2	81.9	73.5	77.5	85.3
$\varepsilon = 40$	80.8	22.2	**93.0**	**15.0**	92.1	26.4	85.9	56.8	83.1	71.4	78.8	84.5
Norm L_2												
PGD2												
$\varepsilon = 0.125$	**71.3**	80.8	67.0	**80.2**	63.9	85.4	56.2	93.8	53.8	94.7	48.6	95.5
$\varepsilon = 0.25$	**63.1**	**83.4**	62.8	86.7	57.1	90.5	52.3	94.6	52.6	94.7	49.9	95.2
$\varepsilon = 0.3125$	64.1	**79.3**	**67.3**	83.1	61.0	88.9	58.0	92.8	57.7	93.3	54.5	94.4
$\varepsilon = 0.5$	72.9	**58.9**	**83.7**	60.7	79.4	73.2	74.6	81.4	73.4	85.4	68.8	90.5
$\varepsilon = 1$	81.0	21.7	**92.9**	**15.5**	91.4	26.4	85.5	57.2	82.9	72.2	78.7	84.7
$\varepsilon = 1.5$	81.5	19.2	**93.2**	**14.2**	91.9	24.2	85.9	56.3	83.2	71.9	79.2	84.4
$\varepsilon = 2$	81.6	19.0	**93.2**	**14.1**	91.9	24.1	85.9	56.3	83.3	71.8	79.2	84.4
DeepFool												
No ε	**91.1**	**22.0**	87.4	33.9	81.9	54.8	70.0	84.4	64.2	91.5	56.3	94.4
CW2												
$\varepsilon = 0.01$	52.9	**90.5**	50.7	90.6	**53.4**	92.2	53.1	94.4	52.0	94.8	50.9	95.0
HOP												
$\varepsilon = 0.1$	**91.3**	**20.9**	89.0	31.0	86.1	49.1	77.0	80.7	72.4	88.1	64.3	92.8
Norm L_∞												
PGDi, FGSM, BIM												
$\varepsilon = 0.03125$	67.2	77.3	77.8	65.2	**82.3**	**59.7**	78.0	72.1	73.7	83.8	64.1	92.2
$\varepsilon = 0.0625$	69.0	83.6	85.3	47.4	**92.0**	**29.6**	90.7	35.7	88.0	45.6	81.3	78.3
$\varepsilon = 0.25$	72.0	67.4	91.8	23.2	**95.9**	**8.8**	94.1	15.4	92.6	19.5	91.6	26.5
$\varepsilon = 0.5$	58.3	84.8	84.2	44.1	**94.6**	**9.7**	91.2	16.5	90.5	18.8	91.3	22.3
PGDi, FGSM, BIM, SA												
$\varepsilon = 0.125$	69.0	79.1	84.1	41.9	**88.9**	**40.8**	86.6	52.3	85.4	60.4	80.7	79.0
PGDi, FGSM, BIM, CWi												
$\varepsilon = 0.3125$	66.6	75.0	**80.6**	**51.5**	80.0	61.1	72.0	84.0	67.2	90.0	60.0	93.6
No norm												
STA												
No ε	84.8	**33.8**	**85.0**	41.5	82.7	52.4	72.9	77.7	70.2	81.7	63.1	92.1

work.

Simulations adversarial attack according to different ε

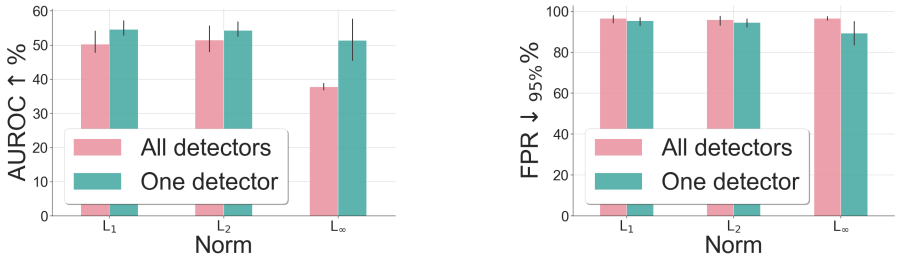
As discussed in Section 3.4, both NSS and the *shallow* detectors aggregated via the proposed method are trained on natural and adversarial examples created with PGD algorithm and L_∞ norm constraint. We show in Tables C.2 to C.5 the results of the two methods according to $\varepsilon \in \{.03125, .0625, .125, .25, .3125, .5\}$.

C.2.2 The proposed aggregator against the adaptive-attacks in the MEAD scenario

We present a new experimental setting to address the case in which also the detectors are attacked at the same time as the target classifier, taking the cue from [BHP+21, CW17a, TCBM20, CW17b]. It is important to note that, in the spirit of the MEAD framework, we are not simply considering a scenario in which

Table C.4: Simultaneous attacks detection: NSS on SVHN. We train NSS on natural and adversarial examples created with PGD algorithm and L_∞ norm constraint. The perturbation magnitude ε is shown in the columns. We indicate in **bold** the best result.

	NSS											
	0.03125		0.0625		0.125		0.25		0.3125		0.5	
	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%
Norm L_1												
PGD1												
$\varepsilon = 5$	37.9	89.3	**40.2**	91.3	37.2	89.2	4.9	35.5	0.3	8.5	0.0	**3.1**
$\varepsilon = 10$	33.7	89.3	**36.9**	91.3	34.6	89.2	6.0	35.5	0.4	8.5	0.0	**3.1**
$\varepsilon = 15$	31.9	89.3	**35.6**	91.3	34.4	89.2	7.6	35.5	0.5	8.5	0.1	**3.1**
$\varepsilon = 20$	31.5	89.3	**36.1**	91.3	35.7	89.2	9.5	35.5	0.6	8.5	0.1	**3.1**
$\varepsilon = 25$	32.8	89.3	37.8	91.3	**38.2**	89.2	11.7	35.5	0.9	8.5	0.1	**3.1**
$\varepsilon = 30$	34.5	89.3	39.8	91.3	**40.6**	89.2	14.1	35.5	1.2	8.5	0.1	**3.1**
$\varepsilon = 40$	37.9	89.3	43.1	91.3	**43.4**	89.0	16.4	35.5	2.2	8.5	0.3	**3.1**
Norm L_2												
PGD2												
$\varepsilon = 0.125$	38.7	89.3	**40.8**	91.3	37.6	89.2	4.7	35.5	0.3	8.5	0.0	**3.1**
$\varepsilon = 0.25$	34.0	89.3	**37.2**	91.3	34.6	89.2	5.4	35.5	0.3	8.5	0.0	**3.1**
$\varepsilon = 0.3125$	32.6	89.3	**36.1**	91.3	34.1	89.2	6.1	35.5	0.4	8.5	0.0	**3.1**
$\varepsilon = 0.5$	31.4	89.3	**35.9**	91.3	35.4	89.2	8.9	35.5	0.5	8.5	0.1	**3.1**
$\varepsilon = 1$	37.4	89.3	42.5	91.3	**42.9**	89.2	16.0	35.5	2.1	8.5	0.3	**3.1**
$\varepsilon = 1.5$	40.0	89.3	46.3	91.3	**46.5**	88.4	17.2	35.5	2.8	8.5	0.6	**3.1**
$\varepsilon = 2$	42.1	89.3	49.8	91.3	**50.5**	88.0	18.7	35.5	3.2	8.5	0.8	**3.1**
DeepFool												
No ε	38.1	89.3	**41.3**	91.3	39.7	89.2	9.2	35.5	0.8	8.5	0.1	**3.1**
CW2												
$\varepsilon = 0.01$	37.9	89.3	**41.0**	91.3	39.5	89.2	9.3	35.5	0.8	8.5	0.1	**3.1**
HOP												
$\varepsilon = 0.1$	66.8	82.3	**67.6**	84.2	60.3	84.6	16.4	35.5	2.7	8.5	0.7	**3.1**
Norm L_∞												
PGDi, FGSM, BIM												
$\varepsilon = 0.03125$	84.1	49.7	**86.3**	46.9	77.5	72.1	22.2	33.2	4.3	8.5	1.2	**3.1**
$\varepsilon = 0.0625$	87.4	**0.2**	**88.9**	0.7	87.5	0.6	33.7	16.8	7.4	6.8	2.5	2.7
$\varepsilon = 0.25$	16.7	89.3	51.6	88.9	**52.0**	85.1	35.4	**0.1**	8.4	**0.1**	3.0	**0.1**
$\varepsilon = 0.5$	4.1	89.3	**46.7**	86.7	46.0	84.6	35.4	**0.1**	8.4	**0.1**	3.0	**0.1**
PGDi, FGSM, BIM, SA												
$\varepsilon = 0.125$	22.8	89.3	32.9	91.3	**43.6**	89.2	30.3	32.7	7.1	8.5	2.5	**3.1**
PGDi, FGSM, BIM, CWi												
$\varepsilon = 0.3125$	4.7	89.3	**41.3**	91.3	40.8	89.2	12.7	35.5	1.7	8.5	0.4	**3.1**
No norm												
STA												
No ε	89.3	**0.0**	**91.2**	0.2	85.9	23.4	19.9	33.5	4.2	8.3	1.4	3.1



(a) Analysis AUROC↑% **(b)** Analysis FPR$\downarrow_{95\%}$%

Figure C.2: Our method against the adaptive-attacks under MEAD. We consider the worst case scenario in Table C.6, i.e., when $\alpha = 0.1$.

a *single* adaptive attack is perpetrated on the classifier and detectors, but rather multiple adaptive attacks are concurrently occurring. We extend the framework to include two main cases: *(i)* for attacks on the classifier and the single detectors individually; *(ii)* for attacks on the classifier and all the detectors simultaneously.

The tables with the complete results are Tables C.6 and C.7, where α is the coefficient that controls the gradient's speed of the attack against the detectors.

Table C.5: Simultaneous attacks detection: the proposed method on SVHN. We train NSS on natural and adversarial examples created with PGD algorithm and L_∞ norm constraint. The perturbation magnitude ε is shown in the columns. We indicate in **bold** the best result.

	Ours											
	0.03125		0.0625		0.125		0.25		0.3125		0.5	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1												
PGD1												
$\varepsilon = 5$	**79.3**	**65.2**	77.4	73.4	76.9	78.9	76.9	79.0	76.7	79.5	74.0	84.4
$\varepsilon = 10$	**74.4**	**65.1**	72.8	73.1	71.9	81.6	73.0	82.5	71.9	84.2	66.9	89.4
$\varepsilon = 15$	76.0	**57.0**	75.7	64.6	75.8	73.1	**78.9**	72.5	77.3	74.7	71.9	84.9
$\varepsilon = 20$	77.3	**48.1**	77.9	54.9	79.2	61.9	**83.6**	60.7	82.2	64.3	77.4	76.9
$\varepsilon = 25$	78.2	**40.9**	79.4	44.4	81.4	49.4	**87.0**	48.6	85.7	52.5	81.4	66.7
$\varepsilon = 30$	78.8	**34.4**	80.4	35.3	83.0	36.6	**89.3**	37.2	88.1	41.6	84.4	53.8
$\varepsilon = 40$	79.7	23.4	81.6	22.4	84.7	20.2	**92.6**	**20.0**	91.1	23.0	87.8	30.5
Norm L_2												
PGD2												
$\varepsilon = 0.125$	**82.2**	**61.7**	80.6	68.4	80.3	72.4	80.2	74.5	80.1	73.5	79.7	75.5
$\varepsilon = 0.25$	**75.7**	**63.6**	74.0	71.7	73.3	80.3	74.0	81.7	72.6	82.8	67.8	89.0
$\varepsilon = 0.3125$	**75.5**	**61.6**	74.3	70.1	73.9	78.4	75.2	79.4	73.9	81.7	70.6	86.7
$\varepsilon = 0.5$	77.2	**50.6**	77.6	57.4	78.6	64.1	**82.5**	64.4	81.2	67.1	76.3	79.5
$\varepsilon = 1$	79.5	25.8	81.3	24.8	84.3	**24.1**	**92.3**	24.7	90.7	27.7	87.1	36.4
$\varepsilon = 1.5$	80.2	19.5	82.2	17.6	85.6	14.3	**94.1**	**7.5**	92.9	8.6	89.9	11.8
$\varepsilon = 2$	80.5	19.4	82.5	17.5	85.9	14.1	**94.9**	**5.3**	94.5	6.8	90.7	9.5
DeepFool												
No ε	**96.3**	**8.6**	95.9	10.5	95.0	12.9	94.9	12.0	95.3	12.1	95.5	12.6
CW2												
$\varepsilon = 0.01$	**59.7**	**76.3**	57.2	80.1	53.4	89.9	54.2	92.0	51.1	93.5	44.3	96.1
HOP												
$\varepsilon = 0.1$	**96.1**	**7.9**	95.6	9.8	95.9	11.7	96.0	10.2	95.9	9.9	**96.1**	10.0
Norm L_∞												
PGDi, FGSM, BIM												
$\varepsilon = 0.03125$	74.3	**60.0**	75.8	60.3	77.8	62.6	**81.4**	64.9	80.1	67.1	76.7	75.5
$\varepsilon = 0.0625$	78.4	36.0	80.3	34.1	83.2	33.8	89.1	**33.3**	**87.9**	34.4	85.7	37.4
$\varepsilon = 0.25$	80.1	19.4	82.1	17.5	85.2	**15.8**	**92.3**	16.4	92.1	16.8	89.6	17.0
$\varepsilon = 0.5$	80.3	19.4	82.3	17.5	85.5	**14.1**	**92.9**	14.4	91.7	15.2	90.1	14.8
PGDi, FGSM, BIM, SA												
$\varepsilon = 0.125$	78.9	29.0	80.8	**28.1**	83.8	28.7	**89.2**	29.1	88.4	28.9	86.8	28.4
PGDi, FGSM, BIM, CWi												
$\varepsilon = 0.3125$	78.7	33.4	80.5	31.9	83.1	34.0	**88.2**	33.1	88.1	31.7	86.7	**31.2**
No norm												
STA												
No ε	**94.7**	**14.5**	93.3	16.8	89.9	23.1	90.2	23.2	91.0	22.4	91.1	22.4

We try many different values $\alpha = \{.1, 1, 5, 10\}$. The case where α is equal to 0 is added for completeness, and it corresponds to the case where only the target classifier is attacked. We report in Fig. C.2 the comparison of the results between case *(i)* and case *(ii)* on CIFAR10 and $\alpha = 0.1$, as this corresponds to the case with the worst performances. As can be seen, the performances of our aggregator improve when the detectors are attacked singularly. This is particularly interesting for the setting we are dealing with. Indeed, our method is not a new supervised adversarial detection method but a framework to aggregate detectors, in this case, applied to the adversarial detection problem. Hence, it does not propose solving the problem of finding a new robust method for adaptive attacks but rather creating a mixture of experts based on the proposed sound mathematical framework. Thus, an attacker to successfully fool our method needs to have the *complete access to all* the underlying detectors and also *an up-to-the-date knowledge of the detectors employed* as the defender can always include a new

Table C.6: The proposed method against the adaptive-attacks under MEAD. In the following setting, we attack each detector and the classifier once at a time. α is the parameter to control the losses.

	CIFAR10									
	$\alpha = 0$		$\alpha = .1$		$\alpha = 1$		$\alpha = 5$		$\alpha = 10$	
	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L_1										
PGD1*										
$\varepsilon = 5$	**62.1**	**87.1**	61.3	88.6	61.2	89.3	63.1	89.2	62.6	91.3
$\varepsilon = 10$	**56.8**	90.6	53.1	94.5	54.4	93.9	60.0	91.0	60.6	91.9
$\varepsilon = 15$	**69.3**	**84.4**	51.5	96.5	54.7	94.6	64.1	88.1	65.7	87.7
$\varepsilon = 20$	**78.7**	**73.1**	53.4	96.8	55.9	94.9	66.7	84.1	69.4	82.7
$\varepsilon = 25$	**87.1**	**50.8**	54.0	97.2	56.7	94.6	67.8	82.7	71.1	79.0
$\varepsilon = 30$	**90.3**	**35.4**	54.5	97.1	56.6	94.4	68.9	81.1	71.9	78.4
$\varepsilon = 40$	**92.1**	**22.7**	54.4	97.0	57.7	93.6	69.4	79.7	72.9	74.2
Norm L_2										
PGD2*										
$\varepsilon = 0.125$	**63.9**	**85.4**	61.4	88.0	62.4	88.8	63.7	88.5	63.9	89.9
$\varepsilon = 0.25$	**57.1**	**90.5**	52.9	94.2	55.0	93.6	60.6	89.7	61.5	90.3
$\varepsilon = 0.3125$	**61.0**	**88.9**	51.6	95.7	54.1	94.7	62.2	87.8	63.7	87.9
$\varepsilon = 0.5$	**79.4**	**73.2**	52.8	96.8	55.3	94.3	66.2	84.6	68.8	81.5
$\varepsilon = 1$	**91.4**	**26.4**	52.7	96.8	57.3	93.4	69.0	78.3	72.1	74.4
$\varepsilon = 1.5$	**91.9**	**24.2**	53.9	96.1	57.9	91.4	70.5	73.7	74.1	68.1
$\varepsilon = 2$	**91.9**	**24.1**	54.6	94.6	59.3	88.5	72.3	67.8	75.6	62.7
Norm L_∞										
PGDi*, FGSM*, BIM*										
$\varepsilon = 0.03125$	**82.3**	**59.7**	45.3	96.2	46.0	96.4	54.5	91.4	57.4	89.3
$\varepsilon = 0.0625$	**92.0**	**29.6**	44.3	96.2	49.8	93.8	59.7	82.4	64.3	76.4
$\varepsilon = 0.5$	**94.6**	**9.7**	62.1	81.3	54.9	81.9	66.1	60.8	68.9	57.9
PGDi*, FGSM*, BIM*, SA										
$\varepsilon = 0.125$	**88.9**	**40.8**	48.6	90.7	54.9	85.0	61.9	73.1	66.3	67.5
PGDi*, FGSM*, BIM*, CWi										
$\varepsilon = 0.3125$	**80.0**	**61.1**	56.6	82.0	56.3	79.6	66.1	66.1	69.2	64.4

detection mechanism to the pool of the detectors.

To give more insights on the proposed aggregator under this setting, we train a *stronger* version of the four shallow detectors where the detectors at training time have seen the corresponding adaptive attacks generated through the PGD algorithm. We report the results in Table C.8 where we focus on the group of simultaneous attacks with L_∞ norm and $\varepsilon = 0.25$ as this represents the worst result of our method in Table C.7. If our method was only good as the best among the detectors, we should expect similar results in Table C.8. In this case, the only solution would be to train a better detector. **However, the strength of the aggregator is not just mimicking the performance of its parts but rather creating a mixture of experts based on the proposed sound mathematical framework.** Therefore, we should expect better performances. Indeed, this consistently holds as the method performs much better than the best detector.

Table C.7: The proposed method against the adaptive-attacks under MEAD. In the following setting, we attack all the detectors and the classifier together at the time. α is the parameter to control the losses.

	CIFAR10									
	$\alpha = 0$		$\alpha = .1$		$\alpha = 1$		$\alpha = 5$		$\alpha = 10$	
	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%	AUROC↑%	FPR$\downarrow_{95\%}$%
Norm L_1										
PGD1*										
$\varepsilon = 5$	**62.1**	**87.1**	61.2	90.4	63.6	86.8	65.8	83.9	66.3	83.2
$\varepsilon = 10$	**56.8**	90.6	50.5	96.4	55.9	91.6	60.1	88.1	61.1	87.2
$\varepsilon = 15$	**69.3**	**84.4**	47.3	97.6	53.8	92.3	62.0	84.9	63.7	83.7
$\varepsilon = 20$	**78.7**	**73.1**	47.1	97.9	54.2	92.5	64.2	82.8	66.8	79.1
$\varepsilon = 25$	**87.1**	**50.8**	47.8	98.0	55.0	92.1	66.5	79.5	68.8	77.2
$\varepsilon = 30$	**90.3**	**35.4**	48.8	98.0	55.8	91.3	67.4	78.5	70.4	75.0
$\varepsilon = 40$	**92.1**	**22.7**	49.1	98.0	56.8	90.5	68.6	77.4	72.5	71.6
Norm L_2										
PGD2*										
$\varepsilon = 0.125$	**63.9**	**85.4**	62.4	88.5	65.0	86.2	66.9	82.9	67.2	81.1
$\varepsilon = 0.25$	**57.1**	**90.5**	51.2	96.0	56.3	91.7	60.6	87.2	61.6	86.8
$\varepsilon = 0.3125$	**61.0**	**88.9**	56.0	94.6	57.9	93.6	65.3	86.4	66.7	86.6
$\varepsilon = 0.5$	**79.4**	**73.2**	46.8	97.8	54.6	91.3	64.5	82.4	66.8	79.5
$\varepsilon = 1$	**91.4**	**26.4**	47.2	98.0	57.8	89.4	69.9	73.8	73.1	71.7
$\varepsilon = 1.5$	**91.9**	**24.2**	47.5	97.6	59.9	86.9	73.2	68.7	76.5	63.1
$\varepsilon = 2$	**91.9**	**24.1**	49.0	97.0	62.8	83.3	75.6	63.7	79.5	56.6
Norm L_∞										
PGDi*, FGSM*, BIM*										
$\varepsilon = 0.03125$	**82.3**	**59.7**	40.2	98.0	47.6	95.5	60.6	86.2	65.0	81.8
$\varepsilon = 0.0625$	**92.0**	**29.6**	37.9	98.0	47.0	95.9	61.9	82.1	65.8	77.1
$\varepsilon = 0.25$	**95.9**	**8.8**	36.5	96.4	47.4	97.7	62.5	92.6	65.4	90.8
$\varepsilon = 0.5$	**94.6**	**9.7**	36.7	96.2	46.0	97.7	61.6	96.1	66.0	94.8
PGDi*, FGSM*, BIM*, SA										
$\varepsilon = 0.125$	**88.9**	**40.8**	38.5	95.9	46.8	95.4	60.1	85.0	61.9	83.2
PGDi*, FGSM*, BIM*, CWi										
$\varepsilon = 0.3125$	**80.0**	**61.1**	37.2	95.3	46.7	97.4	60.9	92.4	64.1	90.1

Table C.8: Comparison between the proposed method and the single detectors (*stronger* version) against the adaptive-attacks. Norm L_∞ and $\varepsilon = 0.25$ (i.e., attacks PGDi*, FGSM*, BIM*).

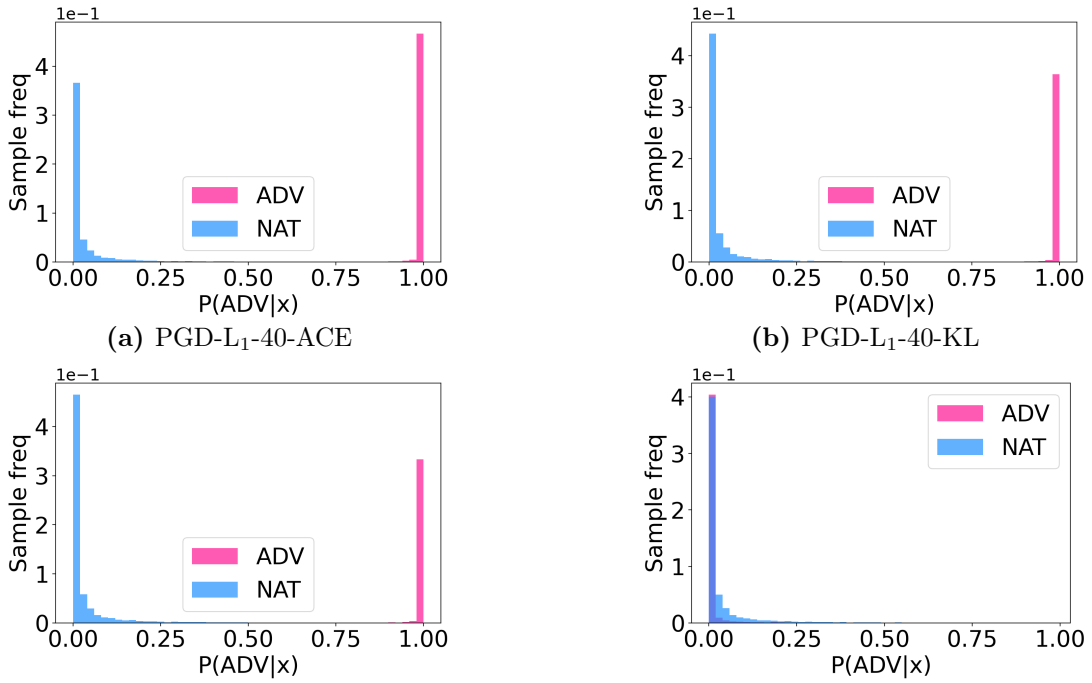
CIFAR10	Ours	ACE	KL	FR	Gini
AUROC↑%	**54.6**	35.7	30.6	26.3	36.2
FPR$\downarrow_{95\%}$%	**73.0**	96.5	97.0	97.4	99.6

C.2.3 AutoAttack

We present an application of AutoAttack [CH20], a state-of-the-art evaluation tool for robustness, redesigned for adversarial detection evaluation and adapted to our simultaneous attacks framework. In its original version, AutoAttack evaluates the accuracy of robust classifiers. In so doing, [CH20] proposes a multiple attacks framework to ensure that at least one attack succeeds in producing an adversarial example for each natural one. In their context, it does not matter which attack will succeed since any successful attack would undermine the accuracy of the target classifier in the same way. In our case, the number of different successful attacks for each natural sample will affect the detection quality since a detector is successful only if it can detect all of them. Because of the above mentioned differences, it is impossible to deploy it directly in our framework without any modifications. A modified version of AutoAttack, adapted to the evaluation of our proposed method, has been implemented, and the results are presented below. While AutoAttack suggests using different attack strategies, in our case, we combine different attack strategies matched with different losses to make the pool of attacks more strong and more diversified.

Table C.9: The proposed method on AutoAttack (MEAD setting). The attacks are APGD-CE, APGD-DLR, FAB, SA.

	CIFAR10	
	Ours	
	AUROC↑%	FPR$_{\downarrow 95\%}$%
Norm L$_1$		
$\varepsilon = 5$	57.1	88.4
$\varepsilon = 10$	67.1	75.7
$\varepsilon = 15$	72.2	66.7
$\varepsilon = 20$	72.7	65.2
$\varepsilon = 25$	72.8	65.6
$\varepsilon = 30$	73.4	64.0
$\varepsilon = 40$	73.6	64.0
Norm L$_2$		
$\varepsilon = 0.125$	67.4	81.0
$\varepsilon = 0.25$	58.0	89.0
$\varepsilon = 0.3125$	58.1	88.8
$\varepsilon = 0.5$	69.4	74.7
$\varepsilon = 1$	75.1	61.6
$\varepsilon = 1.5$	76.1	60.7
$\varepsilon = 2$	76.1	60.5
Norm L$_\infty$		
$\varepsilon = 0.03125$	75.7	61.0
$\varepsilon = 0.0625$	76.0	60.7
$\varepsilon = 0.125$	76.8	60.3
$\varepsilon = 0.25$	76.8	60.0
$\varepsilon = 0.3125$	78.6	57.6
$\varepsilon = 0.5$	76.1	60.3



(a) PGD-L_1-40-ACE

(b) PGD-L_1-40-KL

(c) PGD-L_1-40-FR

(d) PGD-L_1-40-Gini

Figure C.3: In pink the results for the adversarial examples and in blue the ones for the naturals. In this simulation, we consider a subset of the available detectors (ACE, KL, FR). Under each plot, we indicate the tested attack configuration parameters: algorithm-L_p-ε-loss.

C.2.4 Additional plots

The specific shape in the histograms depends on the set of considered detectors. To shed light on this fact, we include the plots in Fig. C.3 in which we consider a subset of the available detectors (ACE, KL, FR). These plots should be compared with the ones in Fig. 5.2.

ECOLE
DOCTORALE

Titre : Sécurisation des Algorithmes d'Apprentissage Automatique

Mots clés : Trustworthy AI, Binary Hypothesis Testing, Détection, Misclassification, Adversarial Examples

Résumé : L'objectif de cette thèse est d'étudier différentes méthodes qui peuvent permettre l'utilisation sécuritaire de technologies de IA.
D'abord, nous devons identifier si la prédiction d'un classificateur devrait (ou ne devrait pas) être fiable afin que il soit possible de l'accepter ou de la rejeter. A cet égard, nous proposons un nouveau détecteur qui approxime le discriminateur le plus puissant (Oracle) basé sur la probabilité d'erreur de classification calculée par rapport à la vraie probabilité postérieure du classificateur. Deux scénarios sont étudiés : Totally Black Box (TBB), où seules les soft-predictions sont disponibles et Partially Black Box (PBB) où la propagation du gradient est autorisée pour effectuer le input pre-processing. Le détecteur proposé peut être appliqué à n'importe quel modèle pre-trained, il ne nécessite pas d'informations préalables sur le dataset et est aussi simple que les méthodes les plus basiques disponibles dans la littérature.
Nous poursuivons en abordant le problème de simultaneous adversarial example detection. Nous proposons un nouveau framework multi-armed pour évaluer les détecteurs sur la base de plusieurs stratégies d'attaques. Parmi celles-ci, nous utilisons trois nouvelles fonctions objectifs pour générer des attaques. La mesure de performance proposée est basée sur le scénario du worst case : la détection est réussie si et seulement si toutes les différentes attaques sont correctement reconnues. De plus, en suivant ce framework nous dérivons formellement une méthode simple mais efficace pour agréger les décisions de plusieurs détecteurs entraînés éventuellement fournis par une tierce partie. Alors que chaque détecteur a tendance à sous-performer ou à échouer dans la détection de types d'attaques qu'il n'a jamais vus au moment de l'entraînement, notre framework permet d'agréger avec succès les connaissances des détecteurs disponibles pour garantir un algorithme de détection robuste. La méthode proposée présente de nombreux avantages : elle est simple car elle ne nécessite pas d'entraînement supplémentaire des détecteurs donnés ; elle est modulaire, permettant aux méthodes existantes (et futures) d'être fusionnées en une seule ; elle est générale car elle peut reconnaître simultanément des exemples adverses créés selon différents algorithmes et objectifs d'entraînement.

Title : Securing Machine Learning Algorithms

Keywords : Trustworthy AI, Binary Hypothesis Testing, Detection, Misclassification, Adversarial Examples

Abstract : This thesis aims to investigate various methods that can enable the safe use of AI technologies.
In the first part, we tackle the problem of identifying whether the prediction of a DNN classifier should (or should not) be trusted so that, consequently, it would be possible to accept or reject it. In this regard, we propose a new detector which approximates the most powerful (Oracle) discriminator based on the probability of classification error with respect to the true class posterior probability. The proposed detector can be applied to any pre-trained model. It does not require prior information about the underlying dataset and is as simple as the simplest available methods in the literature.
We address in the second part the problem of simultaneous adversarial example detection. We propose a novel multi-armed framework for evaluating detectors based on several attack strategies. Among them, we make use of three new objectives to generate attacks. The proposed performance metric is based on the worst-case scenario: detection is successful if and only if all different attacks are correctly recognized. Moreover, following this setting, we formally derive a simple yet effective method to aggregate the decisions of multiple trained detectors, possibly provided by a third party. While every single detector tends to underperform or fail at detecting types of attack that it has never seen at training time, our framework successfully aggregates the knowledge of the available detectors to guarantee a robust detection algorithm. The proposed method has many advantages: it is simple as it does not require further training of the given detectors; it is modular, allowing existing (and future) methods to be merged into a single one; it is general since it can simultaneously recognize adversarial examples created according to different algorithms and training (loss) objectives.

Institut Polytechnique de Paris
91120 Palaiseau, France