Protecting Deep Learning Systems Against Attack : Enhancing Adversarial Robustness and Detection

Marine Picot

université PARIS-SACLAY

McGill

Protecting Deep Learning Systems Against Attack: Enhancing Adversarial Robustness and Detection

Protéger les systèmes de deep learning face aux attaques : Améliorer la robustesse adversaire et la détection

Thèse de doctorat de l'université Paris-Saclay et de McGill University

École doctorale n° 580 Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat : Sciences des réseaux, de l'information et de la communication
Graduate School : Informatique et sciences du numérique. Référent : Faculté des sciences d'Orsay

Thèse préparée dans les unités de recherche **Laboratoire des signaux et systèmes** (Université Paris-Saclay, CNRS, CentraleSupélec) et **Department of Electrical and Computer Engineering** (McGill University, QC, Canada) , sous la direction de **Pablo PIANTANIDA**, Professeur et la co-direction de **Fabrice LABEAU**, Professeur.

Thèse soutenue à Montréal (Québec, Canada), le 7 mars 2023, par

Marine PICOT

Composition du jury

Membres du jury avec voix délibérative

Florence d'ALCHÉ-BUC Professeure, TELECOM Paris	Présidente
Christian DESROSIER Professeur, Département de génie logiciel et des TI, École de Technologie Supérieure (ETS)	Rapporteur
Julien MAIRAL Chargé de recherche (HDR), INRIA	Rapporteur & Examinateur
Ioannis PSAROMILIGKOS Eq. Maitre de Conférence (eq. HDR), Department of Electrical and Computer Engineering, McGill University	Rapporteur & Examinateur
Céline HUDELOT Professeure, MICS, CentraleSupélec	Examinatrice
Deepa KUNDUR Professeure, Electrical & Computer Engineering, University of Toronto	Examinatrice

THESE DE DOCTORAT

NNT: 2023UPASG017

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Protéger les systèmes de deep learning face aux attaques : améliorer la robustesse adversaire et la détection

Mots clés : Deep Learning, IA, Smart Grids, Sécurité, Attaques, Défenses

Résumé : Au cours de la dernière décennie, l'apprentissage profond a été à l'origine de percées dans de nombreux domaines différents, tels que le traitement du langage naturel, la vision par ordinateur et la reconnaissance vocale. Cependant, il est désormais connu que les modèles basés sur l'apprentissage profond sont extrêmement sensibles aux perturbations, en particulier lorsque la perturbation est bien conçue et générée par un agent malveillant. Cette faiblesse des réseaux neuronaux profonds tend à empêcher leur utilisation dans des applications critiques, où des informations sensibles sont disponibles, ou lorsque le système interagit directement avec la vie quotidienne des gens. Dans cette thèse, nous nous concentrons sur la protection des réseaux neuronaux profonds contre les agents malveillants de deux manières principales.

La première méthode vise à protéger un modèle des attaques en augmentant sa robustesse, c'est-à-dire la capacité du modèle à prédire la bonne classe même en cas d'attaques. Nous observons que la sortie d'un réseau neuronal profond forme une variété statistique et que la décision est prise sur cette variété. Nous exploitons cette connaissance en utilisant la mesure de Fisher-Rao, qui calcule la distance géodésique entre deux distributions de probabilité sur la variété statistique auquel elles appartiennent. Nous utilisons la mesure de Fisher-Rao pour régulariser la fonction coût utilisée lors de l'apprentissage et augmenter la robustesse du modèle. Nous adaptons ensuite cette méthode à une autre application critique: les réseaux intelligents (Smart Grids), qui, en raison de divers besoins de la surveillance et de service, reposent sur des composants cybernétiques, tels qu'un estimateur d'état, ce qui les rend sensibles aux attaques. Nous construisons donc des estimateurs d'état robustes en utilisant des autoencodeurs variationnels et l'extension de notre méthode proposée au cas de la régression.

La deuxième méthode sur laquelle nous nous concentrons et qui vise à protéger les modèles basés sur l'apprentissage profond est la détection d'échantillons adverses. En ajoutant un détecteur au modèle, il est possible d'augmenter la fiabilité des décisions prises par les réseaux neuronaux profonds. De multiples méthodes de détection sont disponibles aujourd'hui, mais elles reposent souvent sur un entraînement lourd et des heuristiques ad-hoc. Dans notre travail, nous utilisons des outils statistiques simples appelés les profondeur de données (data-depth) pour construire des méthodes de détection efficaces supervisées (c'est-à-dire que les attaques sont fournies pendant l'entraînement du détecteur) et non supervisées (c'est-à-dire que l'entraînement ne peut s'appuyer que sur des échantillons propres).

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Title: Protecting Deep Learning Systems Against Attack: Enhancing Adversarial Robustness and Detection

Keywords: Deep Learning, AI, Smart Grids, Security, Attacks, Defences

Abstract: Over the last decade, Deep Learning has been the source of breakthroughs in many different fields, such as Natural Language Processing, Computer Vision, and Speech Recognition. However, Deep Learning-based models have now been recognized to be extremely sensitive to perturbations, especially when the perturbation is well-designed and generated by a malicious agent. This weakness of Deep Neural Networks tends to prevent their use in critical applications, where sensitive information is available, or when the system interacts directly with people's everyday life. In this thesis, we focus on protecting Deep Neural Networks against malicious agents in two main ways.

The first method aims at protecting a model from attacks by increasing its robustness, i.e., the ability of the model to predict the right class even under threats. We observe that the output of a Deep Neural Network forms a statistical manifold and that the decision is taken on this manifold. We leverage this knowledge by using the Fisher-Rao measure, which computes the geodesic distance between two probability distributions on the statistical manifold to which they belong. We exploit the Fisher-Rao measure to regularize the training loss to increase the model robustness. We then adapt this method to another critical application: the Smart Grids, which, due to monitoring and various service needs, rely on cyber components, such as a state estimator, making them sensitive to attacks. We, therefore, build robust state estimators using Variational AutoEncoders and the extension of our proposed method to the regression case.

The second method we focus on that intends to protect Deep-Learning-based models is the detection of adversarial samples. By augmenting the model with a detector, it is possible to increase the reliability of decisions made by Deep Neural Networks. Multiple detection methods are available nowadays but often rely on heavy training and ad-hoc heuristics. In our work, we make use of a simple statistical tool called the data-depth to build efficient supervised (i.e., attacks are provided during training) and unsupervised (i.e., training can only rely on clean samples) detection methods.

Protecting Deep Learning Systems Against Attack: Enhancing Adversarial Robustness and Detection

Marine PICOT

Department of Electrical &
Computer Engineering
McGill University
Montréal, Canada.

Laboratoire des Signaux et Systèmes,
CNRS,
Université Paris Saclay,
CentraleSupelec,
Gif-Sur-Yvette, France.

March 2023

A thesis submitted to McGill University in partial fulfilment of the requirements for the degree of PhD.

Abstract

Over the last decade, Deep Learning has been the source of breakthroughs in many different fields, such as Natural Language Processing, Computer Vision, and Speech Recognition. However, Deep Learning-based models have now been recognized to be extremely sensitive to perturbations, especially when the perturbation is well-designed and generated by a malicious agent. This weakness of Deep Neural Networks tends to prevent their use in critical applications, where sensitive information is available, or when the system interacts directly with people's everyday life. In this thesis, we focus on protecting Deep Neural Networks against malicious agents in two main ways.

The first method aims at protecting a model from attacks by increasing its robustness, i.e., the ability of the model to predict the right class even under threats. We observe that the output of a Deep Neural Network forms a statistical manifold and that the decision is taken on this manifold. We leverage this knowledge by using the Fisher-Rao measure, which computes the geodesic distance between two probability distributions on the statistical manifold to which they belong. We exploit the Fisher-Rao measure to regularize the training loss to increase the model robustness. We then adapt this method to another critical application: the Smart Grids, which, due to monitoring and various service needs, rely on cyber components, such as a state estimator, making them sensitive to attacks. We, therefore, build robust state estimators using Variational AutoEncoders and the extension of our proposed method to the regression case.

The second method we focus on that intends to protect Deep-Learning-based models is the detection of adversarial samples. By augmenting the model with a detector, it is possible to increase the reliability of decisions made by Deep Neural Networks. Multiple detection methods are available nowadays but often rely on heavy training and ad-hoc heuristics. In our work, we make use of a simple statistical tool called the data-depth to build efficient supervised (i.e., attacks are provided during training) and unsupervised (i.e., training can only rely on clean samples) detection methods.

Résumé

Au cours de la dernière décennie, l'apprentissage profond a été à l'origine de percées dans de nombreux domaines différents, tels que le traitement du langage naturel, la vision par ordinateur et la reconnaissance vocale. Cependant, il est désormais connu que les modèles basés sur l'apprentissage profond sont extrêmement sensibles aux perturbations, en particulier lorsque la perturbation est bien conçue et générée par un agent malveillant. Cette faiblesse des réseaux neuronaux profonds tend à empêcher leur utilisation dans des applications critiques, où des informations sensibles sont disponibles, ou lorsque le système interagit directement avec la vie quotidienne des gens. Dans cette thèse, nous nous concentrons sur la protection des réseaux neuronaux profonds contre les agents malveillants de deux manières principales.

La première méthode vise à protéger un modèle des attaques en augmentant sa robustesse, c'est-à-dire la capacité du modèle à prédire la bonne classe même en cas d'attaques. Nous observons que la sortie d'un réseau neuronal profond forme une variété statistique et que la décision est prise sur cette variété. Nous exploitons cette connaissance en utilisant la mesure de Fisher-Rao, qui calcule la distance géodésique entre deux distributions de probabilité sur la variété statistique auquel elles appartiennent. Nous utilisons la mesure de Fisher-Rao pour régulariser la fonction coût utilisée lors de l'apprentissage et augmenter la robustesse du modèle. Nous adaptons ensuite cette méthode à une autre application critique : les réseaux intelligents (Smart Grids), qui, en raison de divers besoins de la surveillance et de service, reposent sur des composants cybernétiques, tels qu'un estimateur d'état, ce qui les rend sensibles aux attaques. Nous construisons donc des estimateurs d'état robustes en utilisant des autoencodeurs variationnels et l'extension de notre méthode proposée au cas de la régression.

La deuxième méthode sur laquelle nous nous concentrons et qui vise à protéger les modèles basés sur l'apprentissage profond est la détection d'échantillons adverses. En ajoutant un détecteur au modèle, il est possible d'augmenter la fiabilité des décisions prises par les réseaux neuronaux profonds. De multiples méthodes de détection sont disponibles aujourd'hui, mais elles reposent souvent sur un entraînement lourd et des heuristiques ad-hoc. Dans notre travail, nous utilisons des outils statistiques simples appelés les profondeur de données (data-depth) pour construire

des méthodes de détection efficaces supervisées (c'est-à-dire que les attaques sont fournies pendant l'entraînement du détecteur) et non supervisées (c'est-à-dire que l'entraînement ne peut s'appuyer que sur des échantillons propres).

Contents

List of Figures

List of Tables

List of Abbreviations

AC	**A**lternative **C**urrent
ACE	**A**dversarial **C**ross **E**ntropy
AdaGrad	**Ada**ptive **Grad**ient Descent
Adam	**Ada**ptive **M**omentum
AE	**A**uto**E**ncoder
AGI	**A**rtificial **G**eneral **I**ntelligence
AI	**A**rtificial **I**ntelligence
APGD	**A**uto **P**rojected **G**radient **D**escent
APPROVED	**A** sim**P**le unsu**P**e**R**vised method f**O**r ad**V**ersarial imag**E** **D**etection
AT	**A**dversarial **T**raining
AUROC	**A**rea **U**nder the **R**eceiver **O**perating **C**haracteristic
AWP	**A**dversarial **W**eight **P**erturbation
BCE	**B**oosted **C**ross-**E**ntropy
BIM	**B**asic **I**terative **M**ethod
BPDA	**B**ackward **P**ass **D**ifferentiable **A**pproximation
CE	**C**ross-**E**ntropy
CNN	**C**onvolutional **N**eural **N**etwork
CV	**C**omputer **V**ision
C&W	**C**arlini & **W**agner
DC	**D**irect **C**urrent
DF	**D**eep**F**ool
DL	**D**eep **L**earning
DLR	**D**ifference of **L**ogits **R**atio
DNN	**D**eep **N**eural **N**etworks
FAB	**F**ast **A**daptive **B**oundary
FDIA	**F**alse **D**ata **I**njection **A**ttack
FGSM	**F**ast **G**radient **S**ign **M**ethod
FIRE	**FI**sher-rao **RE**gularizer
FPR	**F**alse **P**ositive **R**ate
FRD	**F**isher-**R**ao **D**istance
FS	**F**eature **S**queezing

GRU	**G**ated **R**ecurent **U**nit
HAMPER	**H**alsp**A**ce **M**ass de**P**th d**E**tector
$\text{HAMPER}_{\text{AA}}$	HAMPER-**A**ttack-**A**ware
$\text{HAMPER}_{\text{BA}}$	HAMPER-**B**lind-to-**A**ttack
HM	**H**alspace-**M**ass depth
HOP	**HOP** skip jump
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IRW	**I**ntegrated **R**ank-**W**eighted depth
JSMA	**J**acobian-based **S**aliency **M**apping **A**ttack
KD-BU	**K**ernel **D**ensity and **B**ayesian **U**ncertainty
KL	**K**ullback-**L**eibler
LID	**L**ocal **I**ntrinsic **D**imensionality
LSTM	**L**ong **S**hort-**T**erm **M**emory
MILP	**M**ixed-**I**nteger **L**inear **P**rogramming
MLP	**M**ulti-**L**ayer **P**erceptron
NIC	**N**etwork **I**nvarient **C**hecking
NLP	**N**atural **L**anguage **P**rocessing
NSS	**N**atural **S**cene **S**tatistics
OOD	**O**ut-**O**f-**D**istribution
PDF	**P**robability **D**ensity **F**unction
PGD	**P**rojected **G**radient **D**escent
RMS Prop.	**R**oot **M**ean **S**quare **Prop**agation
SA	**S**quare **A**ttack
SGD	**S**tochastic **G**radient **D**escent
SOTA	**S**tate-**O**f-**T**he-**A**rt
STA	**S**patial **T**ransformation **A**ttack
VAE	**V**ariational **A**utoEncoder
ViT	**Vi**sion **T**ransformer

Acknowledgments

They are many people I would like to thank for their contribution during my Ph.D., in many different ways.

First, I would like to express the most profound appreciation for my two supervisors, Prof. Fabrice Labeau and Prof. Pablo Piantanida, for their guidance, and trust during those last four years. Their valuable feedback and our discussions were a great part of this doctorate.

I also want to thank the French Ministère de l'Éducation Nationale, de l'Enseignement Supérieur, de la Recherche et de l'Innovation and McGill University for funding my Ph.D.

I am thankful to all my co-authors, Pierre Colombo, Francisco Messina, Federica Granese, Malik Boudiaf, Ismail Ben Ayed, Marco Romanelli, Guillaume Staerman, Nathan Noiry, and Eduardo Gomes, for our multiple discussions and their hard work. I learnt a lot from each of them.

A special thank you to my dear colleagues, Ralph, Pierre, Nicolas, Mickael, Daouda, and Stephano. I do not know how I could have survived those last four years without our breaks.

I also owe a lot to all of my friends, and I would like to give a special thank you to Julie, Charlotte, Pauline, Charles, Eva, and Abdellatif, for always being there for me.

Last but not least, I would like to thank my family, my parents, Vincent and Marylène, and my siblings, Vivien, Paul, Nicolas and Mélanie, for my upbriging and their everlasting support. You helped me more than you can think to achieve this.

Contributions

The work reported in this manuscript was performed by Marine Picot under the supervision of Professor Fabrice Labeau of the Department of Electrical Engineering, McConnell Building of McGill University, Montréal, and Professor Pablo Piantanida with the International Laboratory of Learning Systems (ILLS), together with the Centre National de la Recherche Scientifique, McGill University, the École de Technologie Supérieure, Mila, CentraleSupélec and Paris-Saclay University.

This thesis is manuscript-based and consists of five research manuscripts (Chapter 3, Chapter 4, Chapter 5, Chapter 6 and Chapter 7). Chapter 1, Chapter 2, Chapter 8 consists of the introduction, a review of literature and the conclusion, respectively.

The manuscripts used in this thesis are:

Chapter 3: M. Picot, F. Messina, M. Boudiaf, F. Labeau, I.BenAyed, P.Piantanida. **Adversarial Robustness via Fisher-Rao Regularization**. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*,
DOI:10.1109/TPAMI.2022.3174724, © 2022 IEEE. Reprinted, with permission, from https://ieeexplore.ieee.org/abstract/document/9773978.

Chapter 4: M. Picot, F. Messina, F. Labeau, P. Piantanida. **Robust Autoencoder-based State Estimation in Power Systems**. *2022 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (ISGT)*,
DOI:10.1109/ISGT50606.2022.9817514,
© 2022 IEEE. Reprinted, with permission, from https://ieeexplore.ieee.org/document/9817514.

Chapter 5: M. Picot, F. Messina, F. Labeau, P. Piantanida. **Robust State Estimation Against Adversarial Noise**, submitted at *IEEE Transactions on Smart Grid (TSG)*.

Chapter 6: F. Granese*, M. Picot*, P. Colombo, G. Staerman, M. Romanelli, F. Messina, P. Piantanida. **A Halfspace-Mass Depth-Based Detector for Adversarial Attack Detection**, *Transactions on Machine Learning Research (TMLR)*,
https://openreview.net/pdf?id=YtU0nDb5e8

Chapter 7: M. Picot, G. Staerman, N. Noiry, F. Messina, P. Piantanida, P. Colombo. **A Simple Unsupervised Data Depth-based Method to Detect Adversarial Images**, submitted at *Transactions on Machine Learning Research (TMLR)*.

For the manuscript in Chapter 3, I designed the experiments, developed the code, wrote the manuscript, and helped with the theoretical part. F. Messina and P. Piantanida provided the most significant part of the theoretical part and helped with the writing, while M. Boudiaf provided assistance with the experimental aspect and gave feedback on the manuscript, and I. BenAyed and F. Labeau provided feedback.

For the manuscripts in Chapter 4 and Chapter 5, I designed the experiments, provided the code, and wrote the manuscripts while F. Messina, P. Piantanida, and F. Labeau provided scientific and writing suggestions.

For the manuscript in Chapter 6, G. Staerman provided the tool, P. Colombo, F. Granese and I found the application, F. Granese and I designed the experiments, developed the code, and wrote the paper. P. Colombo and G. Staerman greatly helped with the writing. M. Romanelli, F. Messina, and P. Piantanida provided scientific suggestions and feedback on the writing part.

Finally, for the manuscript in Chapter 7, I developed the experiments and the code and wrote the manuscript. P. Colombo provided scientific and writing suggestions. G. Staerman and N. Noiry, actively helped with the writing, while F. Messina and P. Piantanida provided feedback.

Chapter 1

Introduction

Contents

1.1 General Context

1.1.1 Deep Neural Networks: powerful tools with loopholes

Deep Neural Networks have allowed impressive breakthroughs in different fields, such as Natural Language Processing (NLP) or Computer Vision (CV). Deep Learning (DL)-based systems are more and more used in many applicative domains, as in medicine [IQBAL and collab., 2021], autonomous cars [MOZAFFARI and collab., 2020], bots [LOHOKARE and collab., 2020], or ad recommendations [DU and collab., 2021] among others. Such Artificial Intelligence (AI)-based systems can access critical and personal data about individuals, such as their medical history or credit card information, and their decision can impact our society or directly affect individuals' lives (for example, the autonomous car crashing with passengers inside).

In this context, many concerns about the potential failures of large neural networks, which are not trustworthy enough, limit their adoption. They are due to the poor understanding of the behavior of large neural networks, which are often seen as black boxes. An essential line of research with high industrial and societal impacts consists in designing tools to make them more reliable. These tools are crucial to ensure the wide adoption of DL-based systems. Among the problems of high interest, the undertaken research is expected to take into consideration the following aspects:

- Ensure that no transmission of data to an unauthorized recipient (data leakage) is possible, i.e., **increasing the global system security**.
- If it is impossible, at least ensure that the potentially leaked data does not contain sensitive information, i.e., **increasing data privacy**.
- **Increase the reliability of the deep neural network's decision process**, i.e., finding ways to determine whether the user can trust the decision process.

The primary focus of this thesis is on security aspects. These topics are extensively studied in the research community, as illustrated by the continuous growth in the number of research works addressing security aspects the machine learning which have been published in top-tier AI conferences. In what follows, we provide the reader an overview of some of the main issues that should be addressed to develop safe and trustworthy DL-based systems.

1.1.2 How to ensure no private information leakage

In the privacy-related domain, the main goal is to ensure that no sensitive information is available, even if malicious agents retrieve data. Many different techniques have been studied, and can be summarized in three main categories, each acting

on one of the phase on the DL-based system deployment: modifying the data [ERLINGSSON and collab., 2014; GOLDWASSER and MICALI, 1984; HUANG and collab., 2017; SWEENEY, 2002], acting on the training phase [HESAMIFARD and collab., 2017; IYENGAR and collab., 2019; MOHASSEL and ZHANG, 2017], or using the inference outputs [GILAD-BACHRACH and collab., 2016; MIRESHGHALLAH and collab., 2020a; RIAZI and collab., 2018; WANG and collab., 2018]. We refer the reader to [MIRESHGHALLAH and collab., 2020b] for a more extensive survey about the threats and defensive techniques regarding privacy.

1.1.3 How to increase our trust on DL-based results

Another security-related problem includes measuring and ensuring the reliability of DNNs' decisions. In this field, the goal is to find ways to increase our trust in the final decision made by a DNN. This problem is particularly interesting as large neural networks often perform well when limited to input data whose probability distribution is close to the training dataset on which the models were trained. This can be an essential source of dysfunction in real-world contexts. Indeed, data characteristics are constantly evolving and particularly subject to distributional shifts. At test time, two main scenarios can be distinguished, depending on the source of the input. In the first scenario, we cannot be sure whether the test sample is from the original data source or whether it comes from another environment. We usually call it the *out-of-distribution* (OOD) scenario. In contrast, in the second scenario, we are sure that the input sample is from the original data distribution . However, we do not know how much we can trust the system's decision. We usually call it the *in-distribution* scenario.

How to ensure trust in the environment. Let us consider the case where the system is deployed in a potentially altered environment, i.e., out-of-distribution samples can be fed to the system. Most methods in the OOD literature, to ensure that the input of a deep neural model comes from the original environment, focus on deploying a detector that distinguishes between samples that originate from the original environment and samples that do not. As in the privacy-related fields, OOD detectors can affect different phases of the classifier's deployment procedure. First, the detection method can retrain a classifier for which distinguishing between natural and OOD samples is easier, using techniques from *contrastive training* [HENDRYCKS and collab., 2019; WINKENS and collab., 2020], *regularization* [HEIN and collab., 2019; LEE and collab., 2021; NANDY and collab., 2021], *generative learning* [REN and collab., 2019; SCHLEGL and collab., 2017; VERNEKAR and collab., 2019; XIAO and collab., 2020; ZHANG and collab., 2021] or *ensemble learning* [CHOI and JANG, 2018; VYAS and collab., 2018]). It is also possible for the OOD detection method to only interact with the model outputs. It can be directly at the output layer [HENDRYCKS and

GIMPEL, 2017; HSU and collab., 2020; LIANG and collab., 2018; LIU and collab., 2020], or at different feature levels [GOMES and collab., 2022; KIRICHENKO and collab., 2020; LEE and collab., 2018; QUINTANILHA and collab., 2019; SASTRY and OORE, 2020; ZISSELMAN and TAMAR, 2020].

How to ensure trust in the actual decision. There exists another interesting domain where the goal is to determine whether, in perfect conditions (the system is working in its training environment), we can trust the model to make adequate decisions. One straightforward method is to directly rely on the confidence of the DL model [HENDRYCKS and GIMPEL, 2016], i.e., the higher the confidence is, the lower the uncertainty and the better we can trust the prediction. Since then, different works have tried to develop tools that better catch the relationship between uncertainty and confidence (*MC-dropout* [GAL and GHAHRAMANI, 2016], *Laplace Approximation* [DAXBERGER and collab., 2021], *SWAG* [MADDOX and collab., 2019], *Deep Ensembles* [LAKSHMINARAYANAN and collab., 2017], *DUQ* [VAN AMERSFOORT and collab., 2020], *DOCTOR* [GRANESE and collab., 2021]), or define new confidence scores using auxiliary models (*TrustScore* [JIANG and collab., 2018], *ConfidNet* [CORBIÈRE and collab., 2019]).

1.1.4 How to protect against malicious agents

A third scenario actually exists: what happens if a malicious agent interacts with the system? As a matter of fact, all computational-based systems face the threat of malicious agents that try to disrupt their normal functioning, and cyber-security is a growing field in many communities. It is particularly true in DL as DNNs have been proven to be extremely sensitive to even the slightest well-designed disturbance [SZEGEDY and collab., 2014]. This phenomenon has been shown to be a feature of Deep Neural Networks [ILYAS and collab., 2019]. It is, therefore, crucial to develop techniques to protect the systems against them.

How to handle threats. Tampering with the input can be achieved in multiple ways. One can, for example, add a specific transformation to the input to disrupt a model's behavior. If the transformation is chosen randomly, we talk about *corruption* [HENDRYCKS and DIETTERICH, 2018; SCHNEIDER and collab., 2020]. However, if the tampering is done according to a specific model, under a specific environment, we talk about *adversarial attacks*. Multiple threat methods have been presented over the years [CARLINI and WAGNER, 2017; MADRY and collab., 2018; PAPERNOT and collab., 2016c; SZEGEDY and collab., 2014]. They are extremely powerful at fooling a DNN, and protecting against them has been a hot topic since 2018. Systems deployed without any defense against those kinds of attacks are bound to fail, which could cause massive issues if the system is used in critical conditions. For example, in 2022, XIE and collab. [2022] successfully attacked a stock prediction model.

1.1.5 Focus of this manuscript

All of the aforementioned issues are of extreme importance if we want to develop trustworthy AI systems.

However, although protecting a system from OOD samples and improving its privacy could be enough in some cases, in other cases, especially in critical systems, the vulnerability of DL-based systems to adversarial attacks would prevent their use. In addition, improving the resilience of deep neural networks against attacks can be seen as a worst-case scenario and, therefore, is the most challenging and demanding task. Finally, studying the effect of attacks on DNNs decisions gives us theoretical insights into their weaknesses, and understanding why a DL-based system is fooled can help us understand how a model learns, which is a necessary step towards Artificial General Intelligence (AGI). For all those reasons, we have decided to focus on protecting DL-based systems against malicious agents.

While a plethora of very efficient and relatively easy methods to craft attacks have been developed over the years, the matter of defending against them is an arduous task, and adequate protection methods are not easily found. We, therefore, decided to focus our work on **how to protect DL-based decision processes in the presence of adversarial attacks**.

In the following, we will present the general threats' goal and the motivation behind our work.

1.2 Interacting with Malicious Agents: Attacking and Defending

1.2.1 Global Context

As previously mentioned, AI-based systems all face the issue of handling action from malicious agents. In this setup, there exist two types of scenarios of interest. The first scenario deals with the generation of threats to fool Deep Neural Networks, i.e. behaving as the malicious agent. The second scenario consists in acting as a defender.

The case of attacking neural networks has been widely studied, ending up with many very effective methods that cause DNN-based systems to completely collapse [Carlini and Wagner, 2017; Croce and Hein, 2020; Moosavi-Dezfooli and collab., 2016].

However, while many methods have also been presented in the defender role, very few techniques are acknowledged to be truly efficient at handling malicious agents [Carlini and Wagner, 2017; Tramer and collab., 2020]. The role of the defender is to make the attacker's role harder. A good defense is resilient to attackers

until the capacity, knowledge, or time required for the attacker to break the defense is beyond reasonable.

Given a set of assumptions (for example, the maximal allowed perturbation), it is possible to build defenses that will not collapse [CARMON and collab., 2019; MADRY and collab., 2018; ZHANG and collab., 2019], but there is still margin for improvement, as the performances are not perfect.

We, therefore, have chosen to focus this thesis on crafting new defenses to counter the action of malicious agents.

1.2.2 Notations and problem definition

Adversarial attacks are known to fool any type of DNN-based systems, as attacks for image segmentation [HENDRIK METZEN and collab., 2017; XIE and collab., 2017], object recognition [XIE and collab., 2017], speech recognition [CISSE and collab., 2017], recurrent neural networks [PAPERNOT and collab., 2016a], reinforcement learning [LIN and collab., 2017], generative models [KOS and collab., 2018], variational autoencoder [TABACOF and collab., 2016], language classification [ALZANTOT and collab., 2018; LI and collab., 2020] have been created. Indeed, there exist plenty of applications sensitive to perturbed inputs. Still, techniques developed for image classification can be applied to any sort of application, such as NLP, image segmentation, or signal processing. In addition, the literature on attacking image classifiers is the most developed among all the possible applications. Therefore, we decided to focus mostly on defending against attacks in the **image classification setting**.

In what follows, we formalize the problem of image classification.

The goal of an image classifier is to predict to which class a given input belongs among a certain number of known classes (see Section 2.1 for a quick overview of the Deep Learning methodology).

Notations. First, let us define two spaces: an input space $\mathcal{X} \subset \mathbb{R}^d$ and a label space $\mathcal{Y} = \{1, \dots, \mathrm{C}\}$. In the classical supervised learning problem, data are composed of a pair $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, where $\mathbf{x}$ is an image, and y is its associated label. The unknown joint data distribution is denoted by $p(\mathbf{x}, y)$.

Let us also define the L-layer classifier $f_{\boldsymbol{\theta}}$ parametrized by $\boldsymbol{\theta} \in \Theta$. Let $f_{\boldsymbol{\theta}}^l : \mathcal{X} \to \mathbb{R}^{d_l}$ with $l \in \{1, \dots, \mathrm{L}\}$, denote the output of the l-th layer of the classifier, where d_l is the dimension of the latent space induced by the l-th layer. The class prediction, i.e., the final decision of the classifier, is obtained from the L-th layer softmax output as follows:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = f_{\boldsymbol{\theta}}^{\mathrm{L}}(\mathbf{x}) \triangleq \underset{c \in \mathcal{Y}}{\arg\max}\, q_{\boldsymbol{\theta}}(c|\mathbf{x}) \text{ with } q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) = \mathrm{softmax}(f_{\boldsymbol{\theta}}^{\mathrm{L}-1}(\mathbf{x})).$$

1.2.3 Attacking a neural network

Context. As mentioned earlier, this thesis focuses on defending against adversarial attacks. We will start by defining the threat objective to better understand key challenges.

A brief history. Adversarial samples are well-designed modifications of a given input that aims at disrupting the functioning of a DNN-based system. In 2014, SZEGEDY and collab. [2014] found that this kind of perturbation, already known in the ML community, also disrupted neural networks. Since then, a variety of methods to generate potent attacks have been developed [CARLINI and WAGNER, 2017; CROCE and HEIN, 2020; GOODFELLOW and collab., 2015; MADRY and collab., 2018; MOOSAVI-DEZFOOLI and collab., 2016; PAPERNOT and collab., 2016c; SZEGEDY and collab., 2014].

Problem Formulation. The adversarial generation problem has been defined as follows [SZEGEDY and collab., 2014]. For a given input $\mathbf{x}$ with associated label y, we want to find an adversarial example $\mathbf{x}'$ according to:

$$\begin{aligned} \mathbf{x}' = \arg\min_{\mathbf{x}'} & \|\mathbf{x}' - \mathbf{x}\|_p \\ \text{s.t. } & f_{\boldsymbol{\theta}}(\mathbf{x}') = t \\ & \mathbf{x}' \in [0,1]^d, \end{aligned} \tag{1.1}$$

where t can be a specific class, i.e., targeted attacks, or any class other than y, i.e., untargeted attacks. The condition $\mathbf{x}' \in [0,1]^d$ enforces that the new created sample is still an image.

Having this weakness exposed, researchers started to try to develop defenses to protect deep neural networks. These defenses can be separated into two main groups: **robust models**, where the goal is to preserve the decision integrity, and **detection methods**, where the goal is to ensure that the input is clean.

1.2.4 Defense 1: protecting DNNs' decision integrity against threats

Context. Following the emergence of the threat domain, researchers started to focus on how to defend against these attacks. The first method that caught the community's eye is crafting robust models, i.e., models which make the right decision even under attack.

A brief history. While presenting the generalization of adversarial samples to DNN models, SZEGEDY and collab. [2014] is also the first to mention that back-feeding malicious samples to train a model can help with dealing with them. GOODFELLOW and collab. [2015] is the first to mention Adversarial Training. Since then, a wide variety of defense mechanisms to increase robustness has been proposed [CARMON

and collab., 2019; MADRY and collab., 2018; PAPERNOT and collab., 2016b; WANG and collab., 2019; ZHANG and collab., 2019], some of which have been proven inefficient [CARLINI and WAGNER, 2017].

Problem Formulation. Adversarial training [GOODFELLOW and collab., 2015] is the most commonly used defense and was the first to be recognized as truly effective. The idea behind this method is as follows. To classify adversarial examples correctly, it is possible to train a classifier on this specific task. At each training iteration, perturbations are generated, and the model learns to classify them with their true associated label, i.e., the label of the original image.

The classifier, therefore, tries to solve the following optimization problem:

$$\min_{\boldsymbol{\theta}\in\Theta} \mathrm{L}(\boldsymbol{\theta}), \tag{1.2}$$

where $\mathrm{L}(\boldsymbol{\theta})$ is the risk function used to train the model.

Any risk function can be used to train the model. In their original paper, GOODFELLOW and collab. [2015] proposed to regularize the natural loss with another computed on the Fast-Gradient-Sign-Method attack. However, using this method has been shown to be inefficient. Later, MADRY and collab. [2018] chose to use the expectation of the classical cross-entropy but applied it to the adversarial sample instead of the natural one, i.e.,

$$\mathrm{L}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{x},y)}\left[\max_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|_p\leq\varepsilon} -\log q_{\boldsymbol{\theta}}(y|\mathbf{x}')\right]. \tag{1.3}$$

This method is the first one that has been proven efficient. Further improvements of the original method have been proposed [CARMON and collab., 2019; WANG and collab., 2019; ZHANG and collab., 2019]. One of them ([ZHANG and collab., 2019]) defines a new risk function based on the trade-off between correctly classifying the clean samples and ensuring natural and adversarial samples are similarly classified. The first part is ensured through the minimization of the classical cross-entropy risk. The second is through the minimization of the Kullback-Leibler divergence between natural and adversarial predictions. The trade-off is controlled using λ, which is a hyperparameter. To summarize,

$$\mathrm{L}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{x},y)}\left[-\log q_{\boldsymbol{\theta}}(y|\mathbf{x})\right] + \lambda\,\mathbb{E}_{p(\mathbf{x})}\left[\max_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|_p\leq\varepsilon} \mathrm{KL}(q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})||q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}'))\right], \tag{1.4}$$

where $\mathrm{KL}(q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})||q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')) = \mathbb{E}_{q_{\boldsymbol{\theta}}(y|\mathbf{x})}\left[\frac{q_{\boldsymbol{\theta}}(y|\mathbf{x})}{q_{\boldsymbol{\theta}}(y|\mathbf{x}')}\right]$ is the Kullback-Leibler (KL) divergence between the clean and the disturbed predictions.

The Kullback-Leibler divergence between two probability distributions p and q is a statistical measure measuring how the two probability distributions differ from one another. In other words, it can be seen as how much uncertainty we can expect from q if we know p.

In the case of adversarial learning, finding the disturbance that maximizes the KL divergence between the natural and the perturbed samples will tend to create samples whose predictions are independent of the clean predictions, while minimizing it will force the model to learn how to find a prediction function keeping the information shared between the clean and the attacked samples.

Given a fixed model (i.e., fixed $\boldsymbol{\theta}$), the family $\{q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) : \mathbf{x} \in \mathscr{X}\}$ forms a statistical manifold. DL-based systems' output space therefore forms a manifold and the final decision of this given model is taken on it. Therefore, we can wonder: **How can we use the internal structure of DNN's output space to improve its robustness ?**

1.2.5 Defense 2: ensuring DNNs' input integrity

Context. The second main type of defense consists in ensuring that the input of the DNN is valid (i.e., is not corrupted). To do so, it is possible to craft detection methods to deploy on top of the model to protect.

A brief history. One of the first methods to detect adversarial samples is BENDALE and BOULT [2016], that only uses the output of the penultimate layer to protect their network. Since then, methods have started to use different mechanisms to detect adversarial samples, from the study of input statistics [KHERCHOUCHE and collab., 2020], to the use of different model outputs [LI and LI, 2017; MA and collab., 2018], to the training of ad-hoc models [GONG and collab., 2017; GROSSE and collab., 2017; MENG and CHEN, 2017], to the analysis of the changes in output when the input is modified [HU and collab., 2019; XU and collab., 2018].

Problem Formulation. Crafting a detector is equivalent to finding a function $g : \mathbb{R}^d \to \{0,1\}$ that will associate to the normal behavior the class 0 and the abnormal one the class 1. In practice, we want to find a dissimilarity function $s : \mathbb{R}^d \to \mathbb{R}$, that will output an abnormality score. The higher the score, the more likely the input is corrupted. Using a thresholding step, we can link g and s for a given input $\mathbf{x}$ and a given threshold γ, thanks to:

$$g(\mathbf{x}) = \mathbb{1}_{s(\mathbf{x})>\gamma} = \begin{cases} 0 & \text{if } s(\mathbf{x}) \leq \gamma \\ 1 & \text{if } s(\mathbf{x}) > \gamma. \end{cases} \tag{1.5}$$

Finding an appropriate scoring function to detect adversarial attacks is still an open question.

Depending on the paradigm, the detection methods can be either *supervised*, i.e., at training time, the detector has information about the attacker, or *unsupervised*, i.e., the detector only has access to the natural training samples. In both cases, detection methods tend to rely on ad-hoc heuristics [FEINMAN and collab., 2017; KHERCHOUCHE and collab., 2020; XU and collab., 2018], or on heavy training [MA

and collab., 2018; MENG and CHEN, 2017; RAGHURAM and collab., 2021].

To bypass these limitations, we tried to answer a simple research question: **How can we craft an efficient and effective detection method based on simple tools ?**

1.3 Contribution and Outlines

During this Ph.D., as mentioned in Section 1.2, we addressed two distinct questions.

- **Q1:** How can we use the internal structure of DNN's output space to improve its robustness?
- **Q2:** How can we craft an efficient and effective detection method based on simple tools?

1.3.1 How can we use the internal structure of DNN's output space to improve its robustness ?

In order to address Q1, we studied the impact of a new information-geometric measure to improve the robustness of neural networks: **the Fisher-Rao measure**.

We investigate the problem of optimizing the trade-off between accuracy and robustness. We derive an explicit characterization of the Fisher-Rao Distance (FRD) based on the information-geometric properties of the soft predictions induced space of the neural classifier and propose a new formulation of adversarial defense, called FIsher-rao REgularizer (FIRE).

In addition, we apply a similar method to another practical domain: the state estimation of Smart Grid systems. Indeed, to ensure the appropriate functioning of the electrical grid, monitoring its behavior is crucial. To perform that, we usually estimate the states of the network thanks to the measurable physical quantities available by using a state estimator [ABUR and EXPOSITO, 2004]. However, this cyber component of the power grid makes it highly vulnerable to attacks, such as False Data Injection Attacks (FDIAs) [LIU and collab., 2011], like in Ukraine in 2015 [ALDERSON and DI PIETRO, 2016] when a cyberattack in the electrical network caused a blackout for several hours. We, therefore, extended FIRE method to craft state estimator robust to attacks.

1.3.2 How can we craft an efficient and effective detection method based on simple tools?

In order to address Q2, we first had to find a simple tool that allows us to distinguish between natural and attacked behaviors.

In the statistical community, there exists a simple tool called the **data-depth**, first introduced by TUKEY [1975], that computes a center-outward ordering of a new sample with respect to a given probability distribution. This can be seen as *how close is this new sample to the examples of a known probability distribution?* Since then, multiple definitions of data-depths have been proposed, including halfspace depth [TUKEY, 1975], the simplicial depth [LIU, 1990], the projection depth [LIU, 1992] or the zonoid depth [KOSHEVOY and MOSLER, 1997]. We refer the reader to [STAERMAN, 2022] for an in-depth explanation of the data-depths.

In our work, we decided to use the data-depths to create new methods to detect adversarial examples, depending on the available setting. We studied the detection capabilities of our proposed method under different threat scenarios and provided insights into why our proposed method outperforms others.

1.3.3 Outlines

The rest of this manuscript is organized as follows.

In Chapter 2, we present the different state-of-the-art methods to craft, defend against and detect adversarial examples.

Next, we present our solutions to increase the trade-off between natural performances and robustness in Part I. It is divided as follows.

- In Chapter 3, we present the Fisher-Rao distance in the case of classification, define the new robust risk based on this information-geometric distance, and experimentally compare it to previously used ones.

- In Chapter 4 and Chapter 5, we extended the method presented in Chapter 3 to the Smart Grid problem. After presenting the state estimation problem, we give a quick review of the available techniques to attack and defend against attacks. Finally, we present our proposed method to defend against attack using the Fisher-Rao distance, first under linear approximation, then in the general case.

In in Part II, we present our proposed data-depths detection methods. It is organized as follows.

- In Chapter 6, we describe our data-depth method in the case of supervised adversarial detection. In this case, the defender can access full or partial knowledge about the threats it will encounter. We show that thanks to particular characteristics of the data-depths, our method can remain efficient against attackers with full knowledge about the deployed defense mechanism.

- In Chapter 7, we propose a fully unsupervised detection method. In this case, the defender has no knowledge whatsoever about whether it will be attacked.

We also show that our approach remains partially efficient against attackers with complete knowledge about the deployed defense mechanism.

Finally, in Chapter 8, we present our concluding remarks, the limitations of our work, and the interesting future work that could be done based on our work.

1.4 List of Publications

1.4.1 Journals

1. **Adversarial Robustness via Fisher-Rao Regularization.** *M. Picot*, F. Messina, M. Boudiaf, F. Labeau, I. BenAyed, P. Piantanida. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

2. **A Halfspace-Mass Depth-Based Detector for Adversarial Attack Detection.** *M. Picot**, F. Granese*, G. Staerman, M. Romanelli, F. Messina, P. Piantanida, P. Colombo. *Transactions on Machine Learning Research (TMLR)*.

1.4.2 Conferences

1. **Robust Autoencoder-based State Estimation in Power Systems.** *M. Picot*, F. Messina, F. Labeau, P. Piantanida. *2022 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (ISGT)*.

2. **Modelling the Uncertainty of a Deep Neural Network Enhances its Adversarial Robustness.** *M. Picot*, P. Piantanida, F. Messina, F. Labeau. *2019 Groupe de Recherche et d'Etudes de Traitement du Signal et des Images Conference (GRETSI)*.

3. **MEAD: A Multi-Armed Approach for Evaluation of Adversarial Examples Detectors.** F. Granese* *M. Picot**, M. Romanelli, F. Messina, P. Piantanida. *2022 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*.

1.4.3 Preprints

1. **Robust State Estimation Against Adversarial Noise.** *M. Picot*, F. Messina, F. Labeau, P. Piantanida. Submitted at *IEEE Transactions on Smart Grid (TSG)*.

2. **A Simple Unsupervised Data Depth-based Method to Detect Adversarial Images.** *M. Picot*, G. Staerman, N. Noiry, F. Messina, P. Piantanida, P. Colombo. Submitted at *Transactions on Machine Learning Research (TMLR 2023)*.

3. **Adversarial Attack Detection Under Realistic Constraints.** *M. Picot*, N. Noiry, P. Piantanida, P. Colombo. Submitted at *Transactions on Machine Learning Research (TMLR 2023)*.

4. **Toward Stronger Textual Attack Detectors.** P. Colombo, *M. Picot*, G. Staerman, N. Noiry, P. Piantanida. Submitted at *Association for Computer Linguistics (ACL 2023)*.

1.5 References

ABUR, A. and A. G. EXPOSITO. 2004, *Power system state estimation: theory and implementation*, CRC press. 34

ALDERSON, D. and R. DI PIETRO. 2016, «Operational technology: Are you vulnerable?», *Governance Directions*, vol. 68, nº 6, p. 339–343. 34

ALZANTOT, M., Y. SHARMA, A. ELGOHARY, B.-J. HO, M. SRIVASTAVA and K.-W. CHANG. 2018, «Generating natural language adversarial examples», *arXiv preprint arXiv:1804.07998*. 30

BENDALE, A. and T. E. BOULT. 2016, «Towards open set deep networks», in *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 1563–1572. 33

CARLINI, N. and D. WAGNER. 2017, «Adversarial examples are not easily detected: Bypassing ten detection methods», in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, ACM, New York, NY, USA, ISBN 978-1-4503-5202-4, p. 3–14, doi: 10.1145/3128572.3140444. 29

CARLINI, N. and D. WAGNER. 2017, «Towards evaluating the robustness of neural networks», in *2017 IEEE Symposium on Security and Privacy (SP)*, ISSN 2375-1207, p. 39–57, doi: 10.1109/SP.2017.49. 28, 29, 31, 32

CARMON, Y., A. RAGHUNATHAN, L. SCHMIDT, J. C. DUCHI and P. S. LIANG. 2019, «Unlabeled data improves adversarial robustness», in *Advances in Neural Information Processing Systems*, p. 11 192–11 203. 30, 31, 32

CHOI, H.-J. and E. JANG. 2018, «Generative ensembles for robust anomaly detection», *ArXiv*, vol. abs/1810.01392. 27

CISSE, M. M., Y. ADI, N. NEVEROVA and J. KESHET. 2017, «Houdini: Fooling deep structured visual and speech recognition models with adversarial examples», *Advances in neural information processing systems*, vol. 30. 30

CORBIÈRE, C., N. THOME, A. BAR-HEN, M. CORD and P. PÉREZ. 2019, «Addressing failure prediction by learning model confidence», *Advances in Neural Information Processing Systems*, vol. 32. 28

CROCE, F. and M. HEIN. 2020, «Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks», in *International Conference on Machine Learning*, PMLR, p. 2206–2216. 29, 31

DAXBERGER, E., A. KRISTIADI, A. IMMER, R. ESCHENHAGEN, M. BAUER and P. HENNIG. 2021, «Laplace redux-effortless bayesian deep learning», *Advances in Neural Information Processing Systems*, vol. 34, p. 20 089–20 103. 28

DU, C., Z. GAO, S. YUAN, L. GAO, Z. LI, Y. ZENG, X. ZHU, J. XU, K. GAI and K.-C. LEE. 2021, «Exploration in online advertising systems with deep uncertainty-aware learning», in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, p. 2792–2801. 26

ERLINGSSON, Ú., V. PIHUR and A. KOROLOVA. 2014, «Rappor: Randomized aggregatable privacy-preserving ordinal response», in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, p. 1054–1067. 27

FEINMAN, R., R. R. CURTIN, S. SHINTRE and A. B. GARDNER. 2017, «Detecting adversarial samples from artifacts», *arXiv preprint arXiv:1703.00410*. 33

GAL, Y. and Z. GHAHRAMANI. 2016, «Dropout as a bayesian approximation: Representing model uncertainty in deep learning», in *international conference on machine learning*, PMLR, p. 1050–1059. 28

GILAD-BACHRACH, R., N. DOWLIN, K. LAINE, K. LAUTER, M. NAEHRIG and J. WERNING. 2016, «Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy», in *International conference on machine learning*, PMLR, p. 201–210. 27

GOLDWASSER, S. and S. MICALI. 1984, «Probabilistic encryption», *Journal of computer and system sciences*, vol. 28, nº 2, p. 270–299. 27

GOMES, E. D. C., F. ALBERGE, P. DUHAMEL and P. PIANTANIDA. 2022, «Igeood: An information geometry approach to out-of-distribution detection», in *International Conference on Learning Representations*. 28

GONG, Z., W. WANG and W.-S. KU. 2017, «Adversarial and clean data are not twins», *arXiv preprint arXiv:1704.04960*. 33

GOODFELLOW, I. J., J. SHLENS and C. SZEGEDY. 2015, «Explaining and harnessing adversarial examples», *International Conference on Learning Representations*. 31, 32

GRANESE, F., M. ROMANELLI, D. GORLA, C. PALAMIDESSI and P. PIANTANIDA. 2021, «Doctor: A simple method for detecting misclassification errors», *Advances in Neural Information Processing Systems*, vol. 34, p. 5669–5681. 28

GROSSE, K., P. MANOHARAN, N. PAPERNOT, M. BACKES and P. MCDANIEL. 2017, «On the (statistical) detection of adversarial examples», *arXiv preprint arXiv:1702.06280*. 33

HEIN, M., M. ANDRIUSHCHENKO and J. BITTERWOLF. 2019, «Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem», *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 41–50. 27

HENDRIK METZEN, J., M. CHAITHANYA KUMAR, T. BROX and V. FISCHER. 2017, «Universal adversarial perturbations against semantic image segmentation», in *Proceedings of the IEEE international conference on computer vision*, p. 2755–2764. 30

HENDRYCKS, D. and T. DIETTERICH. 2018, «Benchmarking neural network robustness to common corruptions and perturbations», in *International Conference on Learning Representations*. 28

HENDRYCKS, D. and K. GIMPEL. 2016, «A baseline for detecting misclassified and out-of-distribution examples in neural networks», *arXiv preprint arXiv:1610.02136*. 28

HENDRYCKS, D. and K. GIMPEL. 2017, «A baseline for detecting misclassified and out-of-distribution examples in neural networks», in *International Conference on Learning Representations*. 27

HENDRYCKS, D., M. MAZEIKA and T. DIETTERICH. 2019, «Deep anomaly detection with outlier exposure», in *International Conference on Learning Representations*. 27

HESAMIFARD, E., H. TAKABI and M. GHASEMI. 2017, «Cryptodl: Deep neural networks over encrypted data», *arXiv preprint arXiv:1711.05189*. 27

HSU, Y.-C., Y. SHEN, H. JIN and Z. KIRA. 2020, «Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data», *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 10 948–10 957. 28

HU, S., T. YU, C. GUO, W.-L. CHAO and K. Q. WEINBERGER. 2019, «A new defense against adversarial images: Turning a weakness into a strength», *Advances in Neural Information Processing Systems*, vol. 32. 33

HUANG, C., P. KAIROUZ, X. CHEN, L. SANKAR and R. RAJAGOPAL. 2017, «Context-aware generative adversarial privacy», *Entropy*, vol. 19, nº 12, p. 656. 27

ILYAS, A., S. SANTURKAR, D. TSIPRAS, L. ENGSTROM, B. TRAN and A. MADRY. 2019, «Adversarial examples are not bugs, they are features», *Advances in neural information processing systems*, vol. 32. 28

IQBAL, S., G. F. SIDDIQUI, A. REHMAN, L. HUSSAIN, T. SABA, U. TARIQ and A. A. ABBASI. 2021, «Prostate cancer detection using deep learning and traditional techniques», *IEEE Access*, vol. 9, p. 27 085–27 100. 26

IYENGAR, R., J. P. NEAR, D. SONG, O. THAKKAR, A. THAKURTA and L. WANG. 2019, «Towards practical differentially private convex optimization», in *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, p. 299–316. 27

JIANG, H., B. KIM, M. GUAN and M. GUPTA. 2018, «To trust or not to trust a classifier», *Advances in neural information processing systems*, vol. 31. 28

KHERCHOUCHE, A., S. A. FEZZA, W. HAMIDOUCHE and O. DÉFORGES. 2020, «Natural scene statistics for detecting adversarial examples in deep neural networks», in *22nd IEEE International Workshop on Multimedia Signal Processing*, IEEE, p. 1–6. 33

KIRICHENKO, P., P. IZMAILOV and A. G. WILSON. 2020, «Why normalizing flows fail to detect out-of-distribution data», in *Advances in Neural Information Processing Systems*, vol. 33, édité par H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan and H. Lin, Curran Associates, Inc., p. 20 578–20 589. 28

KOS, J., I. FISCHER and D. SONG. 2018, «Adversarial examples for generative models», in *2018 ieee security and privacy workshops (spw)*, IEEE, p. 36–42. 30

KOSHEVOY, G. and K. MOSLER. 1997, «Zonoid trimming for multivariate distributions», *The Annals of Statistics*, vol. 25, nº 5, p. 1998–2017. 35

LAKSHMINARAYANAN, B., A. PRITZEL and C. BLUNDELL. 2017, «Simple and scalable predictive uncertainty estimation using deep ensembles», *Advances in neural information processing systems*, vol. 30. 28

LEE, K., K. LEE, H. LEE and J. SHIN. 2018, «A simple unified framework for detecting out-of-distribution samples and adversarial attacks», in *Advances in Neural Information Processing Systems 31*, édité par S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett, Curran Associates, Inc., p. 7167–7177. 28

LEE, S., C. PARK, H. LEE, J. YI, J. LEE and S. YOON. 2021, «Removing undesirable feature contributions using out-of-distribution data», in *International Conference on Learning Representations*. 27

LI, D., Y. ZHANG, H. PENG, L. CHEN, C. BROCKETT, M.-T. SUN and B. DOLAN. 2020, «Contextualized perturbation for textual adversarial attack», *arXiv preprint arXiv:2009.07502*. 30

LI, X. and F. LI. 2017, «Adversarial examples detection in deep networks with convolutional filter statistics», in *Proceedings of the IEEE international conference on computer vision*, p. 5764–5772. 33

LIANG, S., Y. LI and R. SRIKANT. 2018, «Enhancing the reliability of out-of-distribution image detection in neural networks», in *International Conference on Learning Representations*. 28

LIN, Y.-C., Z.-W. HONG, Y.-H. LIAO, M.-L. SHIH, M.-Y. LIU and M. SUN. 2017, «Tactics of adversarial attack on deep reinforcement learning agents», *arXiv preprint arXiv:1703.06748*. 30

LIU, R. Y. 1990, «On a notion of data depth based on random simplices», *The Annals of Statistics*, vol. 18, nº 1, p. 405–414. 35

LIU, R. Y. 1992, *Data Depth and Multivariate Rank Tests*, North-Holland, Amsterdam, p. 279–294. 35

LIU, W., X. WANG, J. OWENS and Y. LI. 2020, «Energy-based out-of-distribution detection», *Advances in Neural Information Processing Systems*. 28

LIU, Y., P. NING and M. K. REITER. 2011, «False data injection attacks against state estimation in electric power grids», *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, nº 1, p. 1–33. 34

LOHOKARE, A., A. SHAH and M. ZYDA. 2020, «Deep learning bot for league of legends», in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, p. 322–324. 26

MA, X., B. LI, Y. WANG, S. M. ERFANI, S. N. R. WIJEWICKREMA, G. SCHOENEBECK, D. SONG, M. E. HOULE and J. BAILEY. 2018, «Characterizing adversarial subspaces using local intrinsic dimensionality», in *6th International Conference on Learning Representations*. 33

MADDOX, W. J., P. IZMAILOV, T. GARIPOV, D. P. VETROV and A. G. WILSON. 2019, «A simple baseline for bayesian uncertainty in deep learning», *Advances in Neural Information Processing Systems*, vol. 32. 28

MADRY, A., A. MAKELOV, L. SCHMIDT, D. TSIPRAS and A. VLADU. 2018, «Towards deep learning models resistant to adversarial attacks», in *International Conference on Learning Representations*. 28, 30, 31, 32

MENG, D. and H. CHEN. 2017, «Magnet: A two-pronged defense against adversarial examples», in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, édité par B. M. Thuraisingham, D. Evans, T. Malkin and D. Xu, ACM, p. 135–147. 33, 34

MIRESHGHALLAH, F., M. TARAM, A. JALALI, A. T. ELTHAKEB, D. TULLSEN and H. ESMAEILZADEH. 2020a, «A principled approach to learning stochastic representations for privacy in deep neural inference», *arXiv preprint arXiv:2003.12154*. 27

MIRESHGHALLAH, F., M. TARAM, P. VEPAKOMMA, A. SINGH, R. RASKAR and H. ESMAEILZADEH. 2020b, «Privacy in deep learning: A survey», *arXiv preprint arXiv:2004.12254*. 27

MOHASSEL, P. and Y. ZHANG. 2017, «Secureml: A system for scalable privacy-preserving machine learning», in *2017 IEEE symposium on security and privacy (SP)*, IEEE, p. 19–38. 27

MOOSAVI-DEZFOOLI, S., A. FAWZI and P. FROSSARD. 2016, «Deepfool: A simple and accurate method to fool deep neural networks», in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN 1063-6919, p. 2574–2582, doi: 10.1109/CVPR.2016.282. 29, 31

MOZAFFARI, S., O. Y. AL-JARRAH, M. DIANATI, P. JENNINGS and A. MOUZAKITIS. 2020, «Deep learning-based vehicle behavior prediction for autonomous driving applications: A review», *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, nº 1, p. 33–47. 26

NANDY, J., W. HSU and M. L. LEE. 2021, «Towards maximizing the representation gap between in-domain & out-of-distribution examples», . 27

PAPERNOT, N., P. MCDANIEL, A. SWAMI and R. HARANG. 2016a, «Crafting adversarial input sequences for recurrent neural networks», in *MILCOM 2016-2016 IEEE Military Communications Conference*, IEEE, p. 49–54. 30

PAPERNOT, N., P. MCDANIEL, X. WU, S. JHA and A. SWAMI. 2016b, «Distillation as a defense to adversarial perturbations against deep neural networks», in *2016 IEEE symposium on security and privacy (SP)*, IEEE, p. 582–597. 32

PAPERNOT, N., P. D. MCDANIEL, S. JHA, M. FREDRIKSON, Z. B. CELIK and A. SWAMI. 2016c, «The limitations of deep learning in adversarial settings», *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, p. 372–387. 28, 31

QUINTANILHA, I. M., R. DE M. E. FILHO, J. LEZAMA, M. DELBRACIO and L. O. NUNES. 2019, «Detecting out-of-distribution samples using low-order deep features statistics», . 28

RAGHURAM, J., V. CHANDRASEKARAN, S. JHA and S. BANERJEE. 2021, «A general framework for detecting anomalous inputs to dnn classifiers», in *International Conference on Machine Learning*, PMLR, p. 8764–8775. 34

REN, J., P. J. LIU, E. FERTIG, J. SNOEK, R. POPLIN, M. DEPRISTO, J. DILLON and B. LAKSHMINARAYANAN. 2019, «Likelihood ratios for out-of-distribution detection», in *Advances in Neural Information Processing Systems*, vol. 32, édité par H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett, Curran Associates, Inc. 27

RIAZI, M. S., C. WEINERT, O. TKACHENKO, E. M. SONGHORI, T. SCHNEIDER and F. KOUSHANFAR. 2018, «Chameleon: A hybrid secure computation framework for machine learning applications», in *Proceedings of the 2018 on Asia conference on computer and communications security*, p. 707–721. 27

SASTRY, C. S. and S. OORE. 2020, «Detecting out-of-distribution examples with Gram matrices», in *Proceedings of the 37th International Conference on Machine Learning*, *Proceedings of Machine Learning Research*, vol. 119, édité par H. D. III and A. Singh, PMLR, p. 8491–8501. 28

SCHLEGL, T., P. SEEBÖCK, S. M. WALDSTEIN, U. SCHMIDT-ERFURTH and G. LANGS. 2017, «Unsupervised anomaly detection with generative adversarial networks to guide marker discovery», . 27

SCHNEIDER, S., E. RUSAK, L. ECK, O. BRINGMANN, W. BRENDEL and M. BETHGE. 2020, «Improving robustness against common corruptions by covariate shift adaptation», *Advances in Neural Information Processing Systems*, vol. 33, p. 11 539–11 551. 28

STAERMAN, G. 2022, *Functional anomaly detection and robust estimation*, thèse de doctorat, Institut polytechnique de Paris. 35

SWEENEY, L. 2002, «k-anonymity: A model for protecting privacy», *International journal of uncertainty, fuzziness and knowledge-based systems*, vol. 10, n° 05, p. 557–570. 27

SZEGEDY, C., W. ZAREMBA, I. SUTSKEVER, J. BRUNA, D. ERHAN, I. GOODFELLOW and R. FERGUS. 2014, «Intriguing properties of neural networks», *International Conference on Learning Representations*. 28, 31

TABACOF, P., J. TAVARES and E. VALLE. 2016, «Adversarial images for variational autoencoders», *arXiv preprint arXiv:1612.00155*. 30

TRAMER, F., N. CARLINI, W. BRENDEL and A. MADRY. 2020, «On adaptive attacks to adversarial example defenses», *Advances in Neural Information Processing Systems*, vol. 33, p. 1633–1645. 29

TUKEY, J. W. 1975, «Mathematics and the picturing of data», in *Proceedings of the International Congress of Mathematicians*, vol. 2, p. 523–531. 35

VAN AMERSFOORT, J., L. SMITH, Y. W. TEH and Y. GAL. 2020, «Uncertainty estimation using a single deep deterministic neural network», in *International conference on machine learning*, PMLR, p. 9690–9700. 28

VERNEKAR, S., A. GAURAV, V. ABDELZAD, T. DENOUDEN, R. SALAY and K. CZARNECKI. 2019, «Out-of-distribution detection in classifiers via generation», in *Neural Information Processing Systems (NeurIPS 2019), Safety and Robustness in Decision Making Workshop*, https://sites.google.com/view/neurips19-safe-robust-workshop, https://sites.google.com/view/neurips19-safe-robust-workshop. 27

VYAS, A., N. JAMMALAMADAKA, X. ZHU, D. DAS, B. KAUL and T. L. WILLKE. 2018, «Out-of-distribution detection using an ensemble of self supervised leave-out classifiers», in *ECCV (8)*, p. 560–574. 27

WANG, J., J. ZHANG, W. BAO, X. ZHU, B. CAO and P. S. YU. 2018, «Not just privacy: Improving performance of private deep learning in mobile cloud», in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, p. 2407–2416. 27

WANG, Y., D. ZOU, J. YI, J. BAILEY, X. MA and Q. GU. 2019, «Improving adversarial robustness requires revisiting misclassified examples», in *International Conference on Learning Representations*. 32

WINKENS, J., R. BUNEL, A. G. ROY, R. STANFORTH, V. NATARAJAN, J. R. LEDSAM, P. MACWILLIAMS, P. KOHLI, A. KARTHIKESALINGAM, S. A. A. KOHL, TAYLAN. CEMGIL, S. M. A. ESLAMI and O. RONNEBERGER. 2020, «Contrastive training for improved out-of-distribution detection», *ArXiv*, vol. abs/2007.05566. 27

XIAO, Z., Q. YAN and Y. AMIT. 2020, «Likelihood regret: An out-of-distribution detection score for variational auto-encoder», in *Advances in Neural Information Processing Systems*, vol. 33, édité par H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan and H. Lin, Curran Associates, Inc., p. 20 685–20 696. 27

XIE, C., J. WANG, Z. ZHANG, Y. ZHOU, L. XIE and A. YUILLE. 2017, «Adversarial examples for semantic segmentation and object detection», in *Proceedings of the IEEE international conference on computer vision*, p. 1369–1378. 30

XIE, Y., D. WANG, P.-Y. CHEN, J. XIONG, S. LIU and S. KOYEJO. 2022, «A word is worth a thousand dollars: Adversarial attack on tweets fools stock prediction», *arXiv preprint arXiv:2205.01094*. 28

XU, W., D. EVANS and Y. QI. 2018, «Feature squeezing: Detecting adversarial examples in deep neural networks», in *25th Annual Network and Distributed System Security Symposium*, The Internet Society. 33

ZHANG, H., Y. YU, J. JIAO, E. P. XING, L. E. GHAOUI and M. I. JORDAN. 2019, «Theoretically principled trade-off between robustness and accuracy», in *International Conference on Machine Learning*, p. 1–11. 30, 32

ZHANG, Y., W. LIU, Z. CHEN, J. WANG, Z. LIU, K. LI and H. WEI. 2021, «Out-of-distribution detection with distance guarantee in deep generative models», . 27

ZISSELMAN, E. and A. TAMAR. 2020, «Deep residual flow for out of distribution detection», in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 28

Chapter 2

Preliminaries

Chapter 2 Abstract

This thesis explores the use of statistical and information-geometric tools to improve Deep Neural Networks security when facing threats. This chapter aims to provide the reader with the necessary background to apprehend the different contributions of this thesis. In this chapter, after a quick review of the Deep Learning background, we present an overview of the current methods to *(i)* attack neural networks, *(ii)* protect the neural network's decisions and *(iii)* detect attacks. We will present in more detail the various methods used to either attack our proposed defenses or to compare ourselves. We first focus on images, and then present the Smart Grid case. Finally, we quickly present the two main tools we use throught this thesis: the Fisher-Rao distance and the data-depths.

Contents

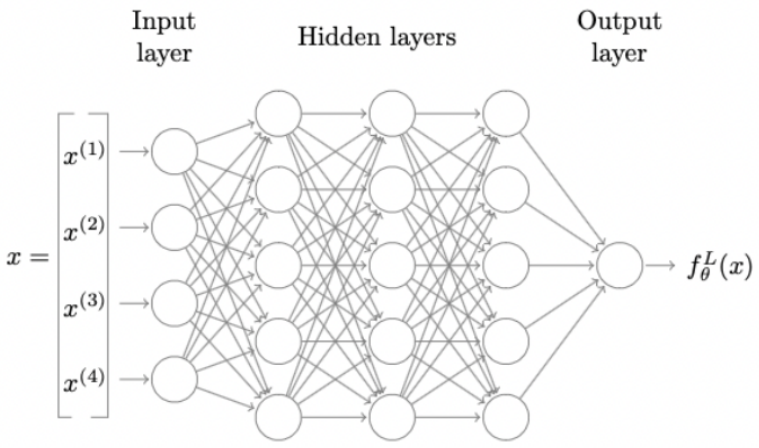


Figure 2.1: Illustration of a Multi-Layer Perceptron

2.1 Deep Learning Background

General supervised models. Let us first recall the previous notations. Let us consider an input space $\mathscr{X} \subset \mathbb{R}^d$, where d is the dimension of the input, and output space $\mathscr{Y}$.We define as $p(\mathbf{x}, y)$ the true joint data distribution.

Let us define a deep neural model $f_{\boldsymbol{\theta}}$ parametrized by $\boldsymbol{\theta} \in \Theta$ with L layers. The output of each layer l is denoted by $f_{\boldsymbol{\theta}}^l(\cdot)$ and belongs in $\mathbb{R}^{d_l}$ where d_l is the output dimension of the layer. Therefore, we have:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = f_{\boldsymbol{\theta}}^{\mathrm{L}}(\mathbf{x}). \tag{2.1}$$

Classification. In the case of supervised classification, the input space $\mathscr{X} \subset \mathbb{R}^d$, and the output space $\mathscr{Y} = \{0, \ldots, \mathrm{C}\}$. In that case, the (L-1)th-layer is called the logits layer, and the Lth layer output the predicted class of a given input. The final decision of the classifier $f_{\boldsymbol{\theta}}$ can be written as follows for a given input x:

$$f_{\boldsymbol{\theta}}^{\mathrm{L}}(\mathbf{x}) = \arg\max_{c \in \mathscr{Y}} q_{\boldsymbol{\theta}}(c|\mathbf{x}) \tag{2.2}$$

where $q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) = \text{softmax}(f_{\boldsymbol{\theta}}^{\text{L}-1}(\mathbf{x}))$ can be seen as soft-probabilities, i.e., each component can be seen as the probability of the input belonging to this specific class.

Specificities of the Computer Vision problem. In the case of CV, the inputs are three-dimensional images, and therefore the input space $\mathcal{X} \subset \mathbb{R}^{w \times h \times c}$ where w and h are the width and the height of the input image, and c is the number of channels (1 for gray-scaled images, 3 for RGB images).

Regression. In the case of supervised regression, the input space $\mathcal{X} \subset \mathbb{R}^d$, and the output space $\mathcal{Y} \subset \mathbb{R}^m$. The main goal of regressors is to estimate a continuous quantity $\mathbf{y} \in \mathcal{Y}$ using observations $\mathbf{x}$.

Architectures.

Multi-Layer Perceptron (MLP). One of the first and most classical neural network architectures is what we call the Multi-Layer Perceptron [HAYKIN, 1994]. It is a feed-forward model, where the information goes only in one way (i.e., from the input to the output). It is composed of at least three layers of aggregated neurons. A layer, composed of multiple neurons, each connected to each neuron of the previous and the next layer, is called a fully-connected layer. Each neuron k in each layer l is characterized by its learnable weights $\mathbf{w}_{l,k} \in \mathbb{R}^{d_{l-1}}$ and biases $\mathbf{b}_{l,k} \in \mathbb{R}$. The relationship between the input of the neuron k in each layer l and its output can be summarized as follows:

$$f_{\boldsymbol{\theta}}^{l,k}(\mathbf{x}) = \sigma(\mathbf{w}_{l,k} \cdot f_{\boldsymbol{\theta}}^{l-1}(\mathbf{x}) + \mathbf{b}_{l,k}), \tag{2.3}$$

where σ is a non-linear function called the activation function (for example, the ReLU function [NAIR and HINTON, 2010]). The global structure of an MLP is presented in Figure 2.1. The output of each layer is then the aggregation of the output of each neuron, i.e., $f_{\boldsymbol{\theta}}^{l}(\mathbf{x}) = [f_{\boldsymbol{\theta}}^{l,1}(\mathbf{x}), f_{\boldsymbol{\theta}}^{l,2}(\mathbf{x}), \ldots, f_{\boldsymbol{\theta}}^{l,d_l-1}(\mathbf{x}), f_{\boldsymbol{\theta}}^{l,d_l}(\mathbf{x})]$.

This type of architecture requires 1-dimensional inputs. To use it in image classification, it would therefore be necessary to transform images into 1-dimensional variables. However, the spatial dependencies of the images would be lost, and for complicated CV tasks, losing such important information could lead to poor results. This is one of the reasons why Convolutional Neural Networks (CNNs) have been considered.

Convolutional Neural Networks. One of the main characteristic of Convolutional Neural Networks (CNNs) [LECUN and collab., 1989] is that they keep spatial dependencies of inputs. They are composed of two main parts. The first one extracts the important features of the input, and the second learns a good way to predict

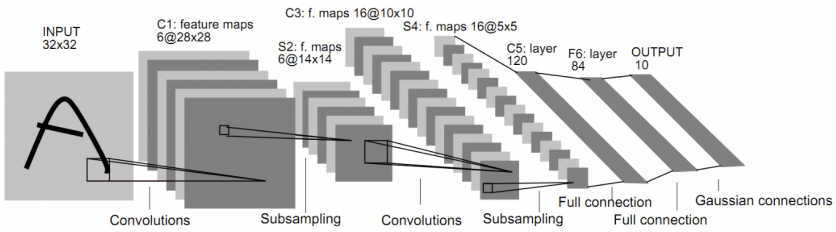


Figure 2.2: CNN structure. More specifically, LeNet's [LeCun and collab., 1989] structure. Source

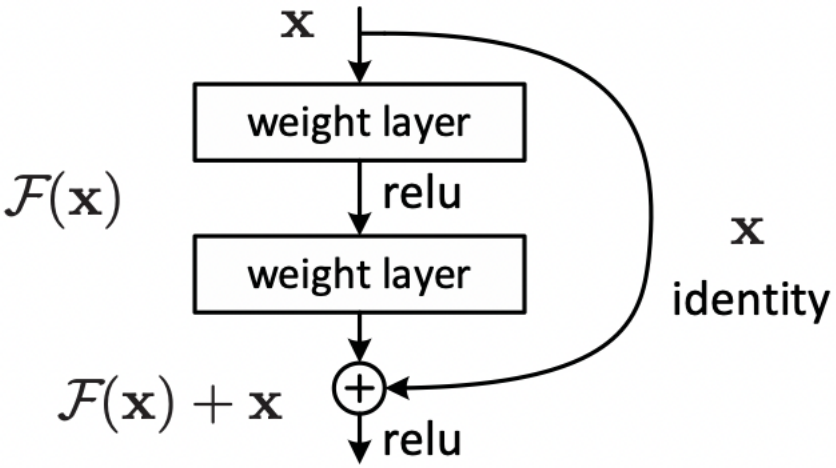


Figure 2.3: Representation of a ResNet residual block, source: [He and collab., 2016], Fig.2

the input's class from those features. While the second part is composed of fully-connected layers, the first one is composed of convolutional layers (see Figure 2.2 for an illustration of CNN's structure). A convolutional layer l is characterized by the filters $\{\mathbf{w}_{l,k}\}_{k\in\{1,\dots,K\}}$ that compose it, their size and number K. The parameters of each filter are learnable, and the output of the layer is the convolution between its input and each of the filters of the layer. We can write the convolutional layer l's operation as:

$$f_{\boldsymbol{\theta}}^{l}(\mathbf{x}) = (f_{\boldsymbol{\theta}}^{l-1} * \mathbf{w}_{l,k})(\mathbf{x}) \tag{2.4}$$

CNNs have gained much importance following the introduction of AlexNet [Krizhevsky and collab., 2012], the first model to vastly outperform previous models on the ImageNet dataset. A few years later the ResNet architecture [He and collab., 2016] was introduced to overcome some of the issue that extremely deep CNN could face during training (vanishing gradients, for example).

ResNets. The ResNet architecture [He and collab., 2016] is one of today's most commonly used architecture for image classification tasks. They introduce a new

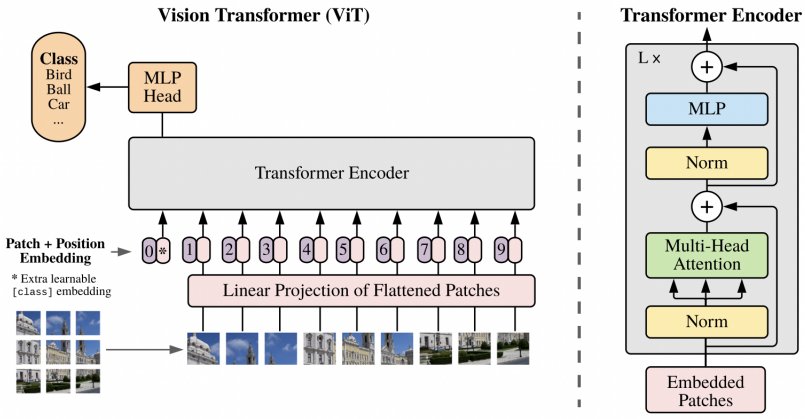


Figure 2.4: Representation of the Vision Transformer's structure. Source: [DOSOVITSKIY and collab., 2020], Fig.1

type of layer called the Residual Block (see Figure 2.3), which not only considers the representation of the input after a few layers but also the original input. Using such architecture allows us to train deeper networks and requires fewer deep parameters than the previous architecture to achieve similar performances.

Transformers. To further improve the image classification results on complicated tasks, the Vision Transformer (ViT) has been introduced. Transformers were initially used in Natural Language Processing applications (NLP) [DEVLIN and collab., 2019]. Transformers were originally introduced for neural machine translation [VASWANI and collab., 2017]. They are composed of multiple transformer cells. The transformer cell is based on self-attention [LIN and collab., 2017] and can be easily parallelized, contrarily to previous used recurrent cells (e.g., LSTM [HOCHREITER and SCHMIDHUBER, 1997], GRU [CHUNG and collab., 2014]) in seq2seq models [CHO and collab., 2014]. The input of a transformer in NLP applications is a sentence, where each word is one of its basic units. In CV applications, the basic unit is the pixel. However, applying the transformer using the pixel as a basic unit is too time and computationally expensive. The first successfully trained Vision Transformer [DOSOVITSKIY and collab., 2020] considered a patch of the input as the basic unit instead of the pixel. They separated their images into 16x16 patches to feed to the model. An overview of the structure of a ViT is presented in Figure 2.4.

Overall, ViTs work as follows. After patching the input image, a linear embedding of each patch is performed, and a position embedding is added. The transformed patches and positions are later fed to the transformer encoder, which is composed of L transformers cells (left side of Figure 2.4 for a detailed presentation of the structure of a transformer cell), which extract the features of the embeddings. After the encoder, as in MLP or CNNs, fully-connected layers are then used to extract useful classification information and output the soft-probabilities $q_{\theta}(\cdot|\mathbf{x})$.

Since their first successful use, ViTs have become the state-of-the-art architecture in classification tasks.

Other architectures. Even though our work mainly focuses on supervised vision classification, we would also like to mention a special case of unsupervised architectures: the AutoEncoders (AE) [BOURLARD and KAMP, 1988]. AE aims at finding a meaningful representation $\mathbf{z}$ of an input $\mathbf{x}$ without supervision. To do so, it will first learn an encoder function $f_{\boldsymbol{\theta}_1}$, where $\mathbf{z} = f_{\boldsymbol{\theta}_1}(\mathbf{x})$ then try to reconstruct $\mathbf{x}$ from $\mathbf{z}$ through the training of a decoder function $f_{\boldsymbol{\theta}_2}$, where $\widehat{\mathbf{x}} = f_{\boldsymbol{\theta}_2}(\mathbf{z})$. A good AutoEncoder will output as $\widehat{\mathbf{x}}$ a good approximation of $\mathbf{x}$ (i.e., $f_{\boldsymbol{\theta}_2}(f_{\boldsymbol{\theta}_1}(\mathbf{x})) \approx \mathbf{x}$) and as $\mathbf{z}$ a meaningful representation of $\mathbf{x}$. The quality of the reconstruction will be controlled by the minimization of a reconstruction loss, while the quality of the representation will be controlled by the minimization of a regularization term. This type of structure is used in many fields, such as data compression [CHENG and collab., 2018; THEIS and collab., 2017], feature extraction [GOGNA and MAJUMDAR, 2019], image denoising [VINCENT and collab., 2008; YASENKO and collab., 2020], for examples.

Variational AutoEncoders [REZENDE and collab., 2014]. Variational AutoEncoder (VAE) is a particular case of AutoEncoders where instead of learning directly the representation $\mathbf{z}$ of $\mathbf{x}$, the model learns the parameters of the probability distribution of the representations. The representation is then sampled from the distributions, and the decoder tries to reconstruct $\mathbf{x}$ from the sampled representation.

The training procedure is key to having reasonably good performing models, whether for classification or regression tasks. The following will present the classical training losses, evaluation metrics, optimizers, and regularization methods used in classical DL problems.

Training procedures. We will first focus on the classical training losses and risks (i.e., the training risk is the expectation over the data of the training loss), which are highly related to the evaluation metrics.

The goals of classifiers and regressors are widely different, therefore, their evaluation metrics are as well. In regression tasks, the goal is to approximate a quantity, usually continuous. An L^p-norm difference between the estimated and the true quantity works well to quantify the performances of a regressor. Depending on the specificities of the problem, L^1 or L^2-norms are usually employed. Defined and differentiable almost everywhere (not at 0 for the L^1-norm), their expectations over the data are both suitable risks to train regression models and evaluate their performances.

Defining suitable training losses and risks is more challenging in classification tasks. The natural way to assess the performances of a given classifier is by counting the number of times it predicts the right class for given inputs. This metric is called

the accuracy (acc.) and can be computed as follows:

$$\text{acc.} = \frac{1}{n_{samples}} \sum_{i=1,\cdots,n_{samples}} \mathbb{1}_{\{f_{\boldsymbol{\theta}}(\mathbf{x}_i)=y_i\}}, \tag{2.5}$$

where y_i is the true label of the input $\mathbf{x}_i$. Yet, this quantity is not suitable as a training risk due to its non-differentiability. The cross-entropy risk has been chosen as a differentiable surrogate of the accuracy. It works efficiently in classification tasks and has interesting connections to other information theory measures. The cross-entropy (CE) risk is defined as, for a given input classifier $f_{\boldsymbol{\theta}}$ with soft-probability distribution $q_{\boldsymbol{\theta}}$:

$$\text{CE}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{x},y)}\big[-\log q_{\boldsymbol{\theta}}(y|\mathbf{x})\big]. \tag{2.6}$$

Once the training risk is chosen, the DL models are trained using classical optimizers, such as the Stochastic Gradient Descent (SGD) optimizer, the Adaptive Gradient Descent (AdaGrad), the Adaptive Momentum (Adam), or the Root Mean Square Propagation (RMS Prop.) optimizer, chosen depending on the specificities of the problem. To avoid overly fitting the training data and generalizing poorly on new incoming input (i.e., overfitting), a variety of techniques can be used. The most widely-spread ones are Dropout (randomly putting input's components, along with some weights, to 0 to force the model to minimize its dependence on specific components) and weight-decay (forcing the weight of each layer to remain small).

Now that we have discussed deploying Deep Neural Networks, we will focus on how to attack them.

2.2 Attacking Neural Networks

Global adversarial problem. SZEGEDY and collab. [2014a] define the adversarial problem as:

$$\begin{aligned} \underset{\mathbf{x}'}{\arg\min} \quad & \|\mathbf{x}' - \mathbf{x}\|_p \\ \text{s.t.} \quad & f_{\boldsymbol{\theta}}(\mathbf{x}') = t \\ & \mathbf{x}' \in [0,1]^d, \end{aligned} \tag{2.7}$$

where t is either the target class (equivalent to targeted attacks) or any class different from the original label y (untargeted attacks). The condition $\mathbf{x}' \in [0,1]^d$ means that we want the adversarial example to represent an image still. In Figure 2.5, the effect of an adversarial example on ImageNet, where the targeted classifier is a GoogLeNet, is represented. This problem is challenging to solve. Therefore, different relaxations

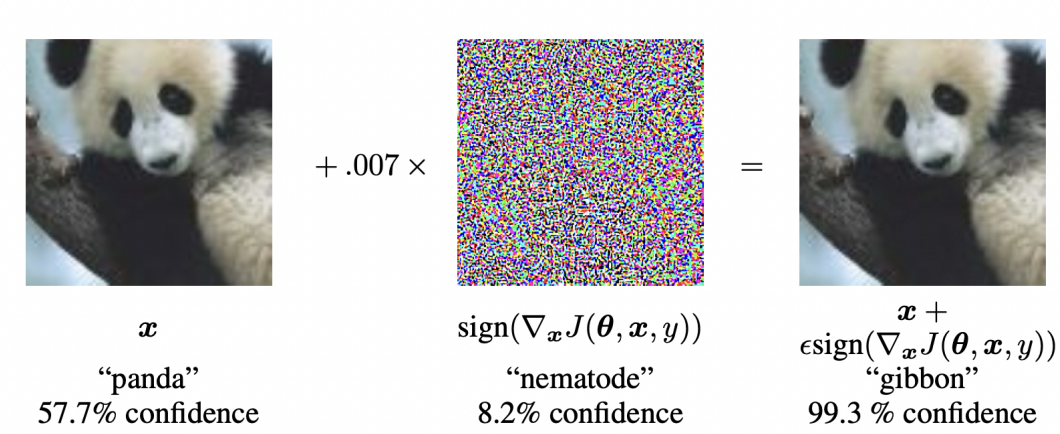


Figure 2.5: Example of the effect of an adversarial sample on ImageNet using a GoogLeNet as the targeted model. Source: [GOODFELLOW and collab., 2015], Fig.1

have been used to derive different methods of adversarial examples generation.

Common relaxation. In their original paper, SZEGEDY and collab. [2014a] proposes the following relaxation.

$$\begin{aligned} &\arg\min_{\mathbf{x}'} && c \cdot \|\mathbf{x}' - \mathbf{x}\|_p + \mathscr{L}(\mathbf{x}, \mathbf{x}', t; \boldsymbol{\theta}) \\ &\text{s.t.} && \mathbf{x}' \in [0,1]^d, \end{aligned} \tag{2.8}$$

where $\mathscr{L}(\mathbf{x}, \mathbf{x}', t; \boldsymbol{\theta})$ is the objective of the attacker, and $c > 0$ is to find. It should be carefully chosen so that minimizing $\mathscr{L}(\mathbf{x}, \mathbf{x}', t; \boldsymbol{\theta})$ will imply that $f_{\boldsymbol{\theta}}(\mathbf{x}') = t$.

Another common relaxation was proposed by MADRY and collab. [2018]:

$$\arg \max_{\mathbf{x}' : \|\mathbf{x}' - x\|_p < \varepsilon} \mathscr{L}(\mathbf{x}, \mathbf{x}', t; \boldsymbol{\theta}). \tag{2.9}$$

In this case, we no longer try to minimize the L^p-norm of the difference between natural and adversarial sample, but we define a maximal allowed perturbation ε given a specific L^p-norm.

Since then, different methods have tried to solve the original problem, choosing a (sometimes modified) version of these relaxations.

Whitebox vs blackbox attacks. Depending on the knowledge about the targeted model the attacker has, we can categorize the adversarial generation methods into two main categories: whitebox and blackbox attacks. In the whitebox setting, the attackers have perfect knowledge about the model. It has access to the training data, the weights and biases, the gradients, and the testing data. However, in the blackbox setting, the attacker has only access to the testing data, sometimes the training dataset, and the final decision of the targeted model. This final decision

can be, for example, the logits, the predictions, or a binary variable stating if the adversarial sample is successful or not.

Many different methods to generate adversarial examples has been proposed since SZEGEDY and collab. [2014b] first acknowledge the problem. Given the huge amount of proposed methods, in whats follows, we will only focus and present the methods to generate adversarial samples we used during this Ph.D to evaluate our proposed methods.

2.2.1 Whitebox attacks

Fast Gradient Sign Method (FGSM). One of the first and simplest method to generate adversarial examples is the Fast Gradient Sign Method (FGSM) [GOODFELLOW and collab., 2015]. The adversarial example $\mathbf{x}'$ is generated thanks to:

$$\mathbf{x}' = \mathbf{x} - \alpha \operatorname{sgn} \nabla_{\mathbf{x}} \mathscr{L}(\mathbf{x}, \mathbf{x}', t; \boldsymbol{\theta}), \text{ (targeted)}, \tag{2.10}$$

$$\mathbf{x}' = \mathbf{x} + \alpha \operatorname{sgn} \nabla_{\mathbf{x}} \mathscr{L}(\mathbf{x}, \mathbf{x}', y; \boldsymbol{\theta}), \text{ (untargeted)}, \tag{2.11}$$

where $\alpha \leq \varepsilon$ is the amplitude of the perturbation, and sgn is the sign function.

This attack scheme is designed to be fast but not necessarily potent.

Projected Gradient Descent. An iterative version of FGSM, called Projected Gradient Descent (PGD) [MADRY and collab., 2018], has been designed such that:

$$\mathbf{x}'^{(0)} = \mathbf{x} + \mathbf{n}, \ \mathbf{n} \sim \text{Unif}([-\varepsilon; \varepsilon]), \tag{2.12}$$

and

$$\mathbf{x}'^{(i+1)} = \mathbf{x}'^{(i)} - \alpha \operatorname{sgn} \nabla_{\mathbf{x}'^{(i)}} \mathscr{L}(\mathbf{x}, \mathbf{x}'^{(i)}, t; \boldsymbol{\theta}), \text{ (targeted)}, \tag{2.13}$$

$$\mathbf{x}'^{(i+1)} = \mathbf{x}'^{(i)} + \alpha \operatorname{sgn} \nabla_{\mathbf{x}'^{(i)}} \mathscr{L}(\mathbf{x}, \mathbf{x}'^{(i)}, y; \boldsymbol{\theta}), \text{ (untargeted)}. \tag{2.14}$$

At each step of the algorithm, the condition $\mathbf{x}' \in [0,1]^d$ is enforced using clipping.

The PGD method has been proven to be both faster and more efficient than the FGSM method. It, therefore, has been widely used. However, PGD has a few issues that can lead to underperforming attacks, as explained in [CROCE and HEIN, 2020b].

Basic Iterative Method. As PGD, the Basic Iterative Method (BIM) [KURAKIN and collab., 2018] is an iterative extension of FGSM. The main difference is that contrary to the initialization of the PGD algorithm, the BIM method starts at the natural sample.

Auto-PGD. According to CROCE and HEIN [2020b], PGD has three main flaws:

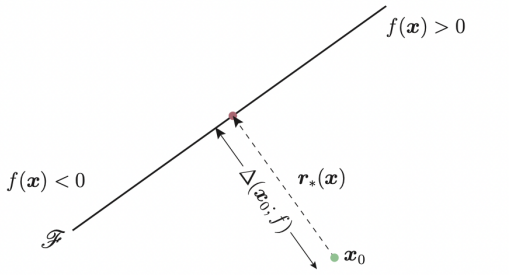


Figure 2.6: DF method for a linear binary classifier. Source: [MOOSAVI-DEZFOOLI and collab., 2016], Fig.2

- The step size is fixed.
- It is agnostic of the budget, i.e., the algorithm does not know how many steps it can perform.
- It is unaware of the trend, i.e., it does not know if the direction it is taking is actually improving the attack objective.

To overcome those issues, they introduced a variant of the PGD algorithm called Auto-PGD (APGD).

It is composed of two phases: one exploration phase, where the algorithm searches for good initial points, and one exploitation phase, where the algorithm tries to maximize the attack strength according to the knowledge accumulated so far.

The APGD method falls to being the PGD algorithm with two main differences. The first one is the computation of the perturbation at each step i:

$$\tilde{\mathbf{x}}'^{(i+1)} = \mathbf{x}'^{(i)} + \eta^{(i)} \nabla_{\mathbf{x}'^{(i)}} \mathscr{L}(\mathbf{x}, \mathbf{x}'^{(i)}, y; \boldsymbol{\theta}), \tag{2.15}$$

$$\mathbf{x}'^{(i+1)} = \mathbf{x}^{(i)} + \alpha * (\tilde{\mathbf{x}}'^{(i+1)} - \mathbf{x}'^{(i)}) + (1-\alpha) * (\mathbf{x}'^{(i)} - \mathbf{x}'^{(i-1)}), \tag{2.16}$$

where $\eta^{(i)}$ corresponds the learned step size at step i. Equation 2.15 is the PGD algorithm, and Equation 2.16 incorporates a momentum with α controlling the influence of the past.

The second difference is that the step size $\eta^{(i)}$ is learnt and no longer fixed. To ensure that the algorithm creates stronger attacks, events are defined during which, if the previous iterations did not improve the attacker objective, the step size is modified, with a restart to the best previous point.

These two modifications of the PGD algorithm tend to create more potent attacks in a similar amount of time.

DeepFool. DeepFool (DF) is a method introduced by MOOSAVI-DEZFOOLI and collab. [2016]. It is an untargeted iterative method based on the gradient of a loss with

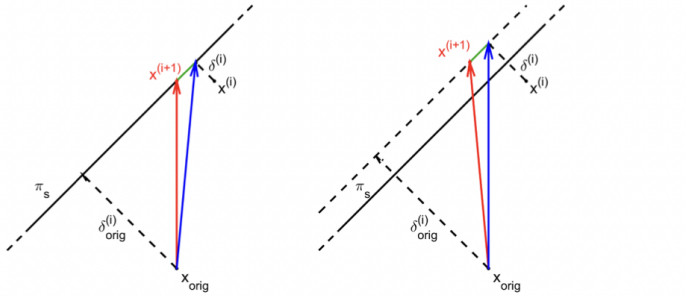


Figure 2.7: FAB method for a linear binary classifier. In blue is the distance between the original example x_{orig} and the projection of the current iteration $x^{(i)}$ onto the hyperplane, in green is the impact of the step towards the original sample, and in red is the new distance. The left figure represents the FAB algorithm using projections onto the hyperplane, while the right one represents the FAB algorithm using projections on the hyperplane plus an extrapolation step (to go over the boundary). Source: [CROCE and HEIN, 2020a], Fig.1

respect to the input. The basic principle of the DF method is presented in Figure 2.6 for a binary classifier.

At each step, until an actual adversarial example is found, they search for the closest decision hyperplane, find the minimal perturbation to add to the sample to project it onto this hyperplane (denoted by $\Delta(x_0; f)$ in Figure 2.6, and add it to the example, creating the new sample (denoted by $r_*(x)$. If the new sample is misclassified, then the search stops. Otherwise, another iteration is done.

Fast Adaptive Boundary. Fast Adaptive Boundary (FAB) was introduced by CROCE and HEIN [2020a]. It is an extension of the DF method, where, in addition to the projection onto the closest hyperplane, they take a step towards the original sample to reduce the needed distortion. An illustration of FAB behavior is available in Figure 2.7. At iteration (i), they compute the distance between the original example x_{orig} and the projection of the example at the current iteration $x^{(i)}$ (in blue in Figure 2.7). Then, they compute a step towards the original point (in green in Figure 2.7) to get the new sample $x^{(i+1)}$. The distance between x_{orig} and $x^{(i+1)}$ is represented in red.

Carlini and Wagner's attack. The Carlini and Wagner's attack (C&W) was introduced by CARLINI and WAGNER [2017]. It can be either a targeted or an untargeted attack. They try to solve the original optimization problem (cf. Equation 2.7).

To that extend, they introduce a function g such that $f_{\boldsymbol{\theta}}(\mathbf{x}') = t$ if and only if $g(\mathbf{x}') \leq 0$. Since the condition can be seen as a minimization problem, we can rewrite the problem as:

$$\min_{\mathbf{x}'} \|\mathbf{x}' - \mathbf{x}\|_p + c\, g(\mathbf{x}') \text{ s.t. } \mathbf{x}' \in [0,1]^d\,, \tag{2.17}$$

where $c \geq 0$ is an hyperparameter.

In their paper, the authors introduce multiple g functions, but the one they recommend for the untargeted case under L^∞-norm constraint is:

$$g(\mathbf{x}) = (\mathbf{z}_y - [\max_{i \neq y} \mathbf{z}_i])^+, \tag{2.18}$$

where $\mathbf{z} = f_{\boldsymbol{\theta}}^{L-1}(\mathbf{x})$, $(\mathbf{z})^+ = \max(0, \mathbf{z})$, $\mathbf{z}_i$ is the i^{th} component of $\mathbf{z}$, and $\mathbf{z}_y$ is the component of $\mathbf{z}$ corresponding to the true class of $\mathbf{x}$. To remove the condition on the norm constraint for $p > 1$, they use the change-of-variable approach with a hyperbolic tangent transformation.

Despite creating potent attacks, the C&W algorithm tends to be highly computationally expensive.

Jacobian-based Saliency Mapping Attack The Jacobian-based Saliency Map Attack (JSMA) was introduced by PAPERNOT and collab. [2016b]. It is a targeted attack. They propose to compute the gradient of each class with respect to each component of the input to extract the sensitivity direction. Then, a saliency map is computed to select the main directions, i.e., the directions in which modifying the pixel will impact the classification the most. The saliency map can be written as:

$$S(\mathbf{x}, t)[i] = \begin{cases} 0, \text{ if } \frac{\partial \mathbf{z}_t}{\partial \mathbf{x}_i}(\mathbf{x}) < 0 \text{ or } \sum_{j \neq t} \frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_i}(\mathbf{x}) > 0, \\ \frac{\partial \mathbf{z}_t}{\partial \mathbf{x}_i}(\mathbf{x}) \left| \sum_{j \neq t} \frac{\partial \mathbf{z}_j}{\partial \mathbf{x}_i}(\mathbf{x}) \right|, \text{ otherwise,} \end{cases}$$

where $\mathbf{z} = f_{\boldsymbol{\theta}}^{L-1}(\mathbf{x})$ is the logit of $\mathbf{x}$ according to the specific model parametrized by $\boldsymbol{\theta}$, $\mathbf{x}_i$ is the i^{th} component of $\mathbf{x}$, $\mathbf{z}_j$ its j^{th} component of $\mathbf{z}$ and $\mathbf{z}_t$ the component of $\mathbf{z}$ corresponding to the targeted class t.

JSMA has been extended to become a blackbox attack [PAPERNOT and collab., 2017] where, instead of crafting the samples on the targeted model, they train a substitute model to attack and use the transferability of neural networks to attack the targeted system.

2.2.2 Blackbox attacks

As previously mentioned, whitebox attackers can create an extremely harmful attack. However, they require much knowledge about the targeted system: they need not only the output of the model but also access to the entire model and its gradients. To bypass this necessity, it is possible to craft attackers that rely on less information. They are called blackbox attackers.

Square Attack. Square Attack (SA) is a blackbox attack introduced by ANDRIUSHCHENKO and collab. [2020]. It is a powerful and rather fast method to generate adversarial samples based only on queries. It is based on randomly selecting

perturbations and adding them to the sample if it decreases the attacker's objective. In this attack, instead of defining pixel-wise perturbations, the attack modifies squares of pixels at the time, whose height decreases over time. The way to sample the perturbation squares differs depending on the L^p-norm constraint considered.

The attacker's objective chosen is the difference between the logits of the true label y and the most likely classes different from y, i.e.,

$$\mathscr{L}(\mathbf{x}, \mathbf{x}', y; \boldsymbol{\theta}) = q_{\boldsymbol{\theta}}(y|\mathbf{x}') - \max_{c \in \mathscr{Y}: c \neq y} q_{\boldsymbol{\theta}}(c|\mathbf{x}'). \tag{2.19}$$

Minimizing such an objective will tend to create samples that are close to the decision boundary.

SA only needs the model's output to run, and it creates highly potent attacks, sometimes even stronger than whitebox attacks.

Spatial Transformation Attack. Spatial Transformation Attack (STA) [ENGSTROM and collab., 2019] is a blackbox attack that relies on finding satisfactory rotations and translations to apply to an input to fool a classifier. Formally, the goal of STA is:

$$\max_{\delta u, \delta v, \phi} \mathscr{L}(\mathbf{x}, \mathbf{x}', y; \boldsymbol{\theta}) \text{ with } \mathbf{x}' = \mathrm{T}(\mathbf{x}; \delta u, \delta v, \phi), \tag{2.20}$$

where $\mathrm{T}(\mathbf{x}; \delta u, \delta v, \phi)$ is the transformation applied to $\mathbf{x}$. The transformation can be written, for a given pixel position (u, v) of the image x, the adversarial new position (u', v') is:

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \delta u \\ \delta v \end{bmatrix}. \tag{2.21}$$

Hop Skip Jump The Hop Skip Jump attack (HOP) [CHEN and collab., 2020] is a blackbox attack based on gradient estimation. In this case, the attack has only access to whether the sample is rightfully or wrongfully classified. The method is iterative, and each iteration i can be divided into four steps:

- Binary search to approach the decision boundary.
- Estimation of the gradient at the decision boundary and computation of the direction of the gradient $\mathbf{v}^{(i)}$.
- Computation of the minimum step size $\eta^{(i)}$.
- $\mathbf{x}'^{(i+1)} = \mathbf{x}'^{(i)} + \eta^{(i)} * \mathbf{v}^{(i)}$.

HOP is the attack presented here that requires the least amount of knowledge about the targeted system to attack.

All of the aforementioned methods are only a small subset of all the possible methods to create adversarial examples. Given the wide variety of choices, **how can we have a common method to evaluate and compare defensive methods ?**

2.2.3 AutoAttack

When the community began to have a huge interest in defensive methods, the question of how to evaluate and compare the methods arose. To answer this question CROCE and HEIN [2020b] introduced AutoAttack.

AutoAttack is a collection of 4 attacks based on the principle of the worst-case scenario.

The four chosen attacks are as follows. First, they use two versions of the APGD algorithm. One where the attacker's objective is the adversarial cross-entropy, and another using the Difference of Logits Ratio (DLR) objective defined as:

$$\mathscr{L}(\mathbf{x}, \mathbf{x}', t; \boldsymbol{\theta}) = \mathrm{DLR}(\mathbf{x}', t) = -\frac{\mathbf{z}'_y - \mathbf{z}'_t}{\mathbf{z}'_{\pi_1} - (\mathbf{z}'_{\pi_3} + \mathbf{z}'_{\pi_4})/2}, \tag{2.22}$$

where $\mathbf{z}' = f_{\boldsymbol{\theta}}^{\mathrm{L}-1}(\mathbf{x}')$ are the logits of $\mathbf{x}'$, $\mathbf{z}'_t$ is the logit of the targeted class, $\mathbf{z}'_y$ is the logit of the original class, and π represents the ordering of the components of $\mathbf{z}'$ in decreasing order. The goal is to push the classification of $\mathbf{x}'$ to t rather than y.

The two final attacks are FAB and SA.

To aggregate the influence of the four attacks, CROCE and HEIN [2020b] proceed as follows. An adversarial example is deemed successful if at least one of the four considered attacks is successful. In other words, the defense is deemed successful for a specific clean input $\mathbf{x}$ if none of the four methods can find a successful adversarial sample based on $\mathbf{x}$ that fools the defense.

Since its introduction in 2020, AutoAttack has become the reference to compare robust methods. A year later, CROCE and collab. [2020] introduced RobustBench to rank the different defensive schemes according to their performances when evaluated using AutoAttack.

It is clear from all of this that there exist plenty of different methods to attack neural networks and that protecting against them is not straightforward. In the following, we will see the main techniques to create defenses to protect DL-based systems against threats.

2.3 Protecting Neural Network's Decisions

Recently, several works focused on improving the robustness of neural networks by investigating various defense mechanisms.

2.3.1 Robustness: different defense mechanisms

Adversarial robustness can be based on plenty of different mechanisms.

Randomness. To build efficient defenses, some papers proposed to leverage randomness. Randomness could be applied at different levels of the system. It is possible to add randomness at the input level [XIE and collab., 2017], at the output of each hidden layer [LIU and collab., 2018a], or directly to the parameters of the model using a Bayesian Neural Network [LIU and collab., 2018b].

Distillation. We could also mention distillation, initially introduced in [HINTON and collab., 2015], and further studied in [PAPERNOT and collab., 2016a] to increase a model's robustness. The idea of distillation is to use a large DNN (the teacher) to train a smaller one (the student), which can perform with similar accuracy while utilizing a temperature parameter to reduce sensitivity to input variations. The resulting defense strategy may be efficient for some attacks but can be defeated with the standard C&W attack.

Adversarial Training. Finally, we want to mention *Adversarial Training*. Adversarial training (AT) was first introduced by GOODFELLOW and collab. [2015]. It is based on augmenting the original data with attacked samples. It is today one of the only defenses that have been proven efficient against adversarial attacks [ATHALYE and collab., 2018]. We will therefore focus on this particular defensive scheme, which is the most popular strategy for enhancing robustness.

Note that, to overcome the lack of guarantees on the task performance beyond standard evaluation metrics, a new line of work called **Certified Robustness** has emerged. The certified defenses aim at training classifiers whose predictions at any input feature will remain constant within a set of neighborhoods around the original input, through different mechanisms: *randomized smoothing* [CAO and GONG, 2017; COHEN and collab., 2019; LECUYER and collab., 2019; LIU and collab., 2018a], *relaxation and duality* [RAGHUNATHAN and collab., 2018b; WONG and collab., 2018], *constraining the global* [CISSE and collab., 2017; GOUK and collab., 2021] or *local* [HEIN and ANDRIUSHCHENKO, 2017] *Lipschitz constant of the model, mixed integer linear programming* [BUNEL and collab., 2018; LOMUSCIO and MAGANTI, 2017], or *adding complementary certification mechanisms* to robust training [RAGHUNATHAN and collab., 2018a; WONG and KOLTER, 2020]. Although these methods are promising, they either do not scale to high-dimensional datasets and models or do not achieve SOTA yet.

2.3.2 Adversarial training

As previously mentioned, Adversarial Training has been introduced by GOODFELLOW and collab. [2015]. It is based on an attack-defense scheme. The attacker aims at creating perturbed inputs by maximizing a loss to fool the classifier, while the

defender's goal is to classify those attacked inputs correctly.

The robust optimization problem solved can be written as follows.

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{p(\mathbf{x},y)} \left[\max_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|_p \leq \varepsilon} \ell(\mathbf{x}', y; \boldsymbol{\theta}) \right], \tag{2.23}$$

where ε denotes the maximal distortion allowed in the adversarial examples according to the L_p-norm, and $\ell(\mathbf{x}, \mathbf{x}', y; \theta)$ is the training loss. Since the exact solution to the above inner max problem is generally intractable, a relaxation is proposed by generating an adversarial example.

The inner attacker design is vital to AT since it has to create meaningful attacks. The chosen mechanism to craft an adversarial sample was the FGSM attack. However, it w quickly shown that, although it provided some successes, it was possible to defeat this defense with stronger attackers [TRAMÈR and collab., 2017].

Later, MADRY and collab. [2018] improved the Adversarial Training framework by modifying the attack to craft adversarial samples. They proposed to no longer use the single-step algorithm FGSM but to use its iterative version, i.e., the PGD algorithm.

An essential choice of the defense mechanism is the robust loss used to attack and defend the network. While GOODFELLOW and collab. [2015] proposed to use a trade-off between the original cross entropy and its adversarial version, i.e., when the input is a perturbed version of the clean one, MADRY and collab. [2018] decided to only use the adversarial cross-entropy, i.e., the cross-entropy between the original label and the corrupted sample.

Considering all the made choices, it is possible to write the optimization problem as the authors in [MADRY and collab., 2018] solved:

$$\min_{\boldsymbol{\theta}} \mathrm{ACE}(\boldsymbol{\theta}), \tag{2.24}$$

where ACE is the Adversarial Cross-Entropy risk,

$$\mathrm{ACE}(\boldsymbol{\theta}) \doteq \mathbb{E}_{p(\mathbf{x},y)} \left[\max_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|_p \leq \varepsilon} -\log q_{\boldsymbol{\theta}}(y|\mathbf{x}') \right]. \tag{2.25}$$

However, it was shown that one way to improve adversarial training is through the choice of this loss.

Improvement based on the loss.

TRADES. ZHANG and collab. [2019] introduces a robustness regularizer based on the Kullback-Leibler divergence. They defined a new risk optimizing a trade-off between natural and adversarial performances, controlled through a hyperparameter λ. The resulting risk is the addition of the natural cross-entropy and the Kullback-

Leibler (KL) divergence between natural and adversarial probability distributions:

$$\mathrm{L}_{\mathrm{TRADES}}(\boldsymbol{\theta}) \doteq \mathbb{E}_{p(\mathbf{x},y)}\left[-\log q_{\boldsymbol{\theta}}(y|\mathbf{x})\right] + \lambda\ \mathbb{E}_{p(\mathbf{x})}\left[\max_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|_p\leq\varepsilon} \mathrm{KL}(q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})\|\, q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}'))\right], \quad (2.26)$$

where

$$\mathrm{KL}(q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})\|\, q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')) \doteq \mathbb{E}_{q_{\boldsymbol{\theta}}(y|\mathbf{x})}\left[\log \frac{q_{\boldsymbol{\theta}}(y|\mathbf{x})}{q_{\boldsymbol{\theta}}(y|\mathbf{x}')}\right]. \quad (2.27)$$

MART. WANG and collab. [2019] uses a robustness regularizer that considers the misclassified inputs and boosted losses. They consider two sets: the set where the original input is rightfully classified and the set where it is misclassified. For each set, they propose a regularizer, equal to $\mathbb{1}\{f_{\boldsymbol{\theta}}(\mathbf{x}') \neq y\} + \mathbb{1}\{f_{\boldsymbol{\theta}}(\mathbf{x}') \neq f_{\boldsymbol{\theta}}(\mathbf{x})\}$, meaning that we want, at the same time, the adversarial example to be rightfully classified, and the natural and adversarial examples to be classified in the same way. Given that, for rightfully classified clean examples, $\mathbb{1}\{f_{\boldsymbol{\theta}}(\mathbf{x}') \neq y\} = \mathbb{1}\{f_{\boldsymbol{\theta}}(\mathbf{x}') \neq f_{\boldsymbol{\theta}}(\mathbf{x})\}$, the total considered risk is:

$$\mathrm{L}_{\mathrm{MART}}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{x},y)}\left[\mathbb{1}\{f_{\boldsymbol{\theta}}(\mathbf{x}') \neq y\} + \lambda\ \mathbb{1}\{f_{\boldsymbol{\theta}}(\mathbf{x}) \neq y\} \cdot \mathbb{1}\{f_{\boldsymbol{\theta}}(\mathbf{x}') \neq f_{\boldsymbol{\theta}}(\mathbf{x})\}\right], \quad (2.28)$$

where λ is a hyperparameter controlling the trade-off between rightfully classifying the adversarial examples and classifying the natural and adversarial examples in the same way.

Since the indicator's function is not differentiable, they relax the three terms as follows. The first one - $\mathbb{1}\{f_{\boldsymbol{\theta}}(\mathbf{x}') \neq y\}$ - is approximated as the boosted cross-entropy(BCE), which is equal to:

$$\mathrm{BCE}(\mathbf{x}', y; \boldsymbol{\theta}) = -\log q_{\boldsymbol{\theta}}(y|\mathbf{x}') - \log(1 - \max_{c\neq y} q_{\boldsymbol{\theta}}(c|\mathbf{x}')). \quad (2.29)$$

The second term - $\mathbb{1}\{f_{\boldsymbol{\theta}}(\mathbf{x}') \neq f_{\boldsymbol{\theta}}(\mathbf{x})\}$ - is approximated as the Kullback-Leibler divergence (Eq. (2.27)), i.e. $\mathbb{1}\{f_{\boldsymbol{\theta}}(\mathbf{x}') \neq f_{\boldsymbol{\theta}}(\mathbf{x})\} \approx \mathrm{KL}(q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})\|\, q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}'))$.

Finally, $\mathbb{1}\{f_{\boldsymbol{\theta}}(\mathbf{x}) \neq y\}$ is approximated by $1 - q_{\boldsymbol{\theta}}(y|\mathbf{x})$, since it will be large for misclassified inputs and small for rightfully classified inputs.

So, the MART risk can be written as:

$$\mathrm{L}_{\mathrm{MART}}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{x},y)}[\max_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|_p\leq\varepsilon} \mathrm{BCE}(\mathbf{x}', y; \boldsymbol{\theta}) + \lambda\ (1 - q_{\boldsymbol{\theta}}(y|\mathbf{x})) \cdot \mathrm{KL}(q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})\|\, q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}'))]. \quad (2.30)$$

Improvement based on additional data.

Whether it is Madry's version of Adversarial Training, TRADES, or MART, they all have been improved in different ways. One of the most successful improvements uses unlabeled data to enhance generalization [ALAYRAC and collab., 2019; CARMON and collab., 2019]. The theoretical argument of this data augmentation is that ad-

versarial generalization requires more examples than natural generalization, i.e., it requires a larger dataset at training time.

To improve robustness, CARMON and collab. [2019] proposes the use of additional data when training on CIFAR-10. Specifically, they use 500k additional images from 80M-TI [1]. Those images have been selected such that their L_2-distance to images from CIFAR-10 are below a fixed threshold. They then use a classifier to predict the pseudo-label of each new image before adding them to the dataset. Finally, they use this augmented dataset to perform adversarial training using the TRADES loss.

ALAYRAC and collab. [2019] propose two methods. The first one is similar to Carmon et al.'s method, but they consider the sum of natural and Adversarial Cross-Entropy weighted by a hyperparameter λ as the loss. The other method does not use pseudo labels. For unlabeled data, only the smoothness of the loss with respect to the adversarial examples is considered. In other words, they consider a natural loss computed only on natural - labeled - examples from the original CIFAR dataset, and for all examples - both labeled and unlabeled - the considered loss is the Kullback-Leibler divergence between natural and adversarial probability distributions, i.e., $\mathrm{KL}(q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) \| q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}'))$. The total sum is a weighted sum of the natural and the adversarial losses.

Other improvements

Other types of improvement have been studied in recent years. Pretraining [HENDRYCKS and collab., 2019], early stopping [RICE and collab., 2020], curriculum learning [ATZMON and collab., 2019], adaptative models [HUANG and collab., 2020], or additional perturbations on the model weights [WU and collab., 2020] are a few examples of these improvements.

It should be noted that the main disadvantage of adversarial training-based methods remains the required computational expenses.

2.4 Ensuring the Input's Integrity

Inspired by the concept of *rejection channels* [CHOW, 1957], which was proposed over 70 years ago for the character recognition problem, another way to protect against adversarial attacks is to construct a detector-based rejection strategy. Its objective is to discriminate malicious samples from clean ones, which implies discarding samples detected as adversarial. Research in this area focuses on both *supervised* and *unsupervised* approaches [ALDAHDOOH and collab., 2021b].

Supervised Detection. The supervised approaches rely on features from natu-

[1] Images available at https://github.com/yaircarmon/semisup-adv

ral and attacked examples generated according to one or more attacks [FEINMAN and collab., 2017; KHERCHOUCHE and collab., 2020; MA and collab., 2018]. The extracted features can be computed either directly on the images [KHERCHOUCHE and collab., 2020] or extracted at the targeted network's layer [CARRARA and collab., 2018; LEE and collab., 2018; LU and collab., 2017; METZEN and collab., 2017]. The supervised detection methods can rely on statistical characteristics linking them to in-training or out-of-training distributions/manifolds [FEINMAN and collab., 2017; GROSSE and collab., 2017; LI and LI, 2017; MA and collab., 2018]). All these methods depend on the defender's knowledge of the threats it will face.

Unsupervised Detection. Knowledge about the attacker is not always available to the defender. To overcome this lack of information, unsupervised detection methods do not rely on prior knowledge of attacks and only learn from clean data at the time of training [MENG and CHEN, 2017; XU and collab., 2018]. Different techniques are used to extract meaningful features. LIANG and collab. [2021]; XU and collab. [2018] rely on *feature squeezing*, MENG and CHEN [2017] relies on the training of an *denoising autoencoder*, MA and collab. [2019] relies on a network invariance, while ZHENG and HONG [2018] uses an auxiliary model.

Novel training procedure. While all the aforementioned methods are deployed on top of an existing model to protect, it is also possible to develop novel training procedures, as *reverse cross-entropy* [PANG and collab., 2018] or *the rejection option* [ALDAHDOOH and collab., 2021a; SOTGIU and collab., 2020].

In our work, we decided to focus on methods that do not modify the underlying classifier to protect. In the following, we will present the detection methods we will compare our work with in more details.

2.4.1 Supervised detection methods

As previously mentioned, supervised detection methods rely on the knowledge and availability of adversarial samples. They can be separated into two categories: the *attack-aware* methods, where one detector is trained per specific attack, and the *blind-to-attack* methods, where a single detector is trained to detect all the threats.

Local Intrinsic Dimensionality (LID). The *Local Intrinsic Dimensionality* method (LID) [MA and collab., 2018] is based on the intuition that adversarial examples lie outside of the clean data manifold. By computing the Local Intrinsic Dimensionality, it is possible to check whether the new point is close to the original data manifold or from another one. The estimation of LID is computed as the inverse of the mean of the log of the distance between a given point to its k nearest neighbors. If a given point is similar to the training distribution, then its distance to either of its k nearest neighbors will always be quite close, and the LID approximate will therefore be close to 0. However, if a sample is not quite similar to the training distribution, its distance

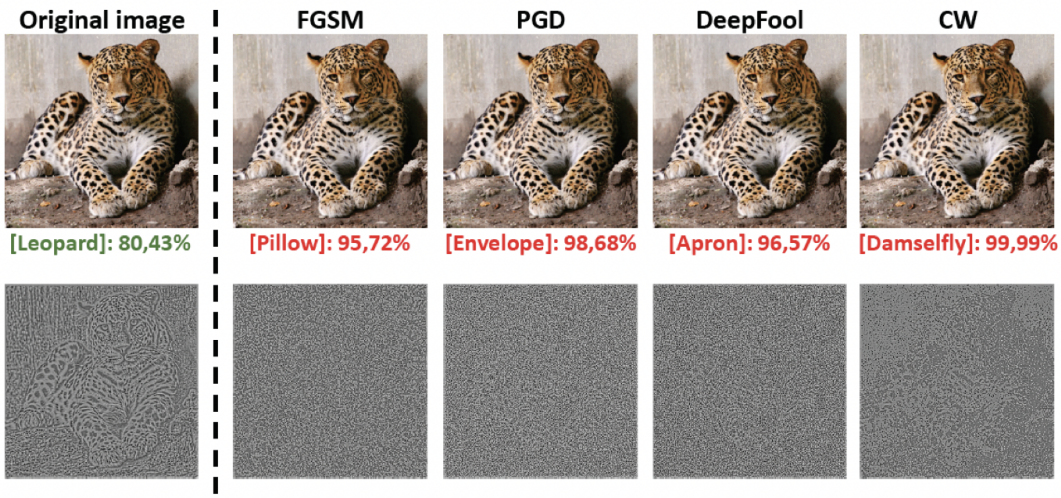


Figure 2.8: Natural Scene Statistics extraction on a natural sample, and various adversarial ones. Source: [KHERCHOUCHE and collab., 2020], Fig.1a

.

to at least one of its nearest neighbors will likely be big, and the LID approximate will increase.

Following this idea, in practice, three versions of each natural sample are used: the clean one, a noisy version, and an attacked one. The LID approximate for each of those points to the clean distribution will be computed at the output of each layer. Those variables will then be used to train a detector that will distinguish between adversarial and normal samples (either clean or noisy samples).

Each strategy to craft adversarial samples can have very different LID characteristics, a detector per type of threat is therefore necessary. LID therefore lies in the *attack-aware* category.

Kernel Density and Bayesian Uncertainty (KD-BU). The *Kernel Density and Bayesian Uncertainty* (KD-BU) method [FEINMAN and collab., 2017] also relies on the idea that adversarial samples lie off the original data manifold. To detect adversarial samples, they first perform a kernel density estimate at the last hidden layer level to detect samples that are far from the original manifold. Then, a bayesian uncertainty estimate is computed to detect when points lie in low-confidence regions of the input space. Both of those characteristics are later used to train a detector that distinguishes between natural and adversarial examples. Once again, the kernel density estimates and the bayesian uncertainty values for different types of attacks can differ a lot, therefore, this method has been created to be *attack-aware*.

Natural Scene Statistics (NSS). The *Natural Scene Statistics* method [KHERCHOUCHE and collab., 2020] relies on the extraction of the natural scene statistics at the image level. Natural scene statistics are statistics that will be very different

for natural and attacked images. Indeed, applying the natural scene statistics will output an image with meaning for clean images. However, for attacked samples, the output image will have no meaning. In Figure 2.8 is presented an example of the extraction of natural scene statistics on a natural sample and various adversarial ones, where we clearly see that, while the NSS image still represents a leopard, the ones for the adversarial examples do not. The Natural Scene Statistics extraction is then used to train a detector to distinguish between natural and attacked samples.

To overcome the need to have a specific detector per attack, NSS decided to train their detector using the natural scene statistics of various attacks. It therefore lies in the *blind-to-attack* category.

2.4.2 Unsupervised detection methods

Feature Squeezing (FS) [Xu and collab., 2018]. The key idea of the *Feature Squeezing* (FS) method is to compare the model's prediction of the original sample with its prediction of the sample after multiple squeezing. The further away they are, the more likely the input is adversarial. In practice, four versions of the input are needed: the original input, a low-precision version, a median-filtered version, and a denoising-filtered version. One inference on the model is required for each of the four inputs. Later, the maximal L_1 difference between the original prediction and each of the other three is picked. FS is, therefore, parameter-free and does not require training.

MagNet [Meng and Chen, 2017]. *MagNet* is based on the idea that adversarial examples do not lie on the data manifold. It uses a detector and a reformer to detect adversarial examples. The detector, an autoencoder trained to reconstruct rightfully natural samples, will try to reject examples that are far from the natural manifold. For a new input, they look at the reconstruction error. If it is small, then the example is clean, else it is adversarial.

The reformer, also an autoencoder, will, given an input, try to find an approximation of it that is on or close to the original manifold. This projection is then fed to the underlying classifier to estimate the class.

The training of the two necessary autoencoders makes this method rather long and computationally and memory expensive to deploy but extremely fast to test.

2.5 Review of the Smart Grid Case

In 2015, the United Nation organization edited 17 goals to change our world. Amongst those goals, goal 7 aims at ensuring access to affordable, reliable, sustainable and modern energy for all. We would need to widely modify our grid's structure to include

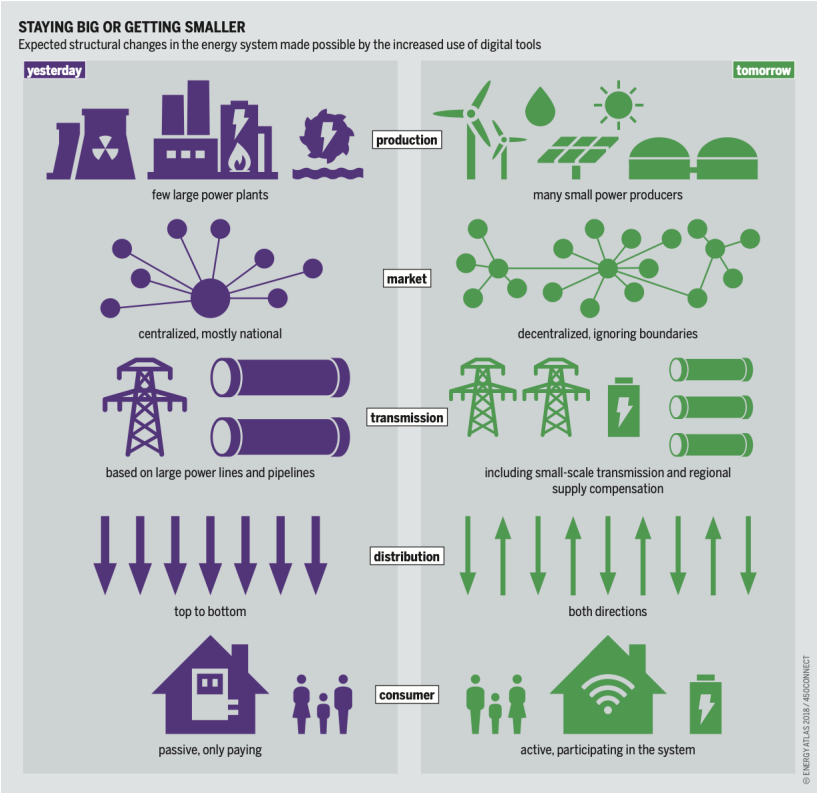


Figure 2.9: Our current grid' structure needs to change to fulfill UN goals to change our world. Source: [HEINRICH, 2018], p.33.

more sustainable energy. Smart Grids are a therefore a crucial tool to achieve this goal.

However, smart grid systems can be sensitive to attacks, and more specifically to False Data Injection Attacks (FDIAs) [LIU and collab., 2011].

In the following, we will provide a quick review of Smart Grid systems, and more specifically of the state estimation problem, how to attack it and how to protect it against attacks.

2.5.1 The state estimation problem

Let $\mathbf{x} = |\mathbf{x}|e^{i \arg \mathbf{x}} \in \mathbb{C}^n$ be the state vector (latent variables) of the power grid, assumed to be random with a prior probability density function (pdf) $p(\mathbf{x})$, and $\mathbf{y} = |\mathbf{y}|e^{i \arg \mathbf{y}} \in \mathbb{C}^m$ the measurement vector. In general, these are related through the following nonlinear equation:

$$\mathbf{y} = \boldsymbol{h}(\mathbf{x}) + \mathbf{z}, \tag{2.31}$$

where $\boldsymbol{h}$ is the measurement function (dependent on the grid topology, line impedances, etc.) and $\mathbf{z} \in \mathbb{C}^m$ is additive noise. This is called the AC model in the literature [ABUR and EXPOSITO, 2004; GIANNAKIS and collab., 2013].

Usually, under certain assumptions about the grid and its operating point, the

problem can be linearized as follows:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}, \tag{2.32}$$

where $\mathbf{H} \in \mathbb{C}^{m \times n}$ is the linearized Jacobian measurement matrix. This is called the DC model.

We define a state estimator as a function $f_{\phi} : \mathbb{C}^m \to \mathbb{C}^n$ such that $\hat{\mathbf{x}} = f_{\phi}(\mathbf{y})$.

State estimators are crucial components for the functioning of smart grid systems, but they are known to be sensitive to faulty samples. Bad data detectors are usually implemented to overcome this issue. They are a-posteriori detectors based on the l_p-norm (where typically $p = 2$) of the residual vector

$$\mathbf{r} = \mathbf{y} - \boldsymbol{h}(\hat{\mathbf{x}}). \tag{2.33}$$

Concretely, the sample is detected as faulty when the $\|\mathbf{r}\|_p > \tau$, where τ is a threshold chosen appropriately to control the trade-off between missed and false detections. Nevertheless, bad data detectors are insufficient to overcome the problem of well-designed attacks. In particular, a theoretical bound on the maximal number of meters that one can attack in order to remain undetected is derived in JIN and collab. [2018].

2.5.2 Attacking and defending state estimator

One of the most critical types of cyber-attacks are FDIAs. In LIU and collab. [2011], the authors present various methods to generate such corrupted samples to bypass the bad data detector under the DC assumption. Let us consider

$$\mathbf{y_a} = \mathbf{y} + \mathbf{a}, \tag{2.34}$$

where $\mathbf{a}$ is the attack vector and $\mathbf{y_a}$ the attacked observation. It has been showed that, to remain undetected, the $\mathbf{a}$ should be a combination of the column vectors of $\mathbf{H}$, i.e., $\mathbf{a} = \mathbf{H}\mathbf{c}$, for all $\mathbf{c}$. The specific values for the vector $\mathbf{c}$ is imposed by the considered scenario, i.e. the goal of the attack.

Most of the research has been done according to the DC model assumption [DÁN and SANDBERG, 2010; KOSUT and collab., 2010; LIU and collab., 2011; PASQUALETTI and collab., 2011; SANDBERG and collab., 2010; XIE and collab., 2010; YUAN and collab., 2011], which leads to a simple linear model for the measurements as a function of the state. LIANG and collab. [2016] present a comprehensive review of the security problem under the DC-model assumption.

Crafting an FDIA under the AC model (Equation 2.31) is a more complex problem. A few works have focused on it, as [HUG and GIAMPAPA, 2012; JIN and collab., 2018; KEKATOS and collab., 2017; LIANG and collab., 2014; TEIXEIRA and collab., 2015; ZHU

and GIANNAKIS, 2012], which considers the nonlinear relation between the state and the measurements. Joint cyber and physical attacks have also been studied [SOLTAN and collab., 2016], leading to the study of detection and recovery of information from line failure under DC and AC assumption [SOLTAN and collab., 2018; SOLTAN and ZUSSMAN, 2018].

JIN and collab. [2018] presented the attacker's optimization problem as:

$$\begin{aligned} \max_{\mathbf{x_a},\mathbf{a}} \quad & g(\mathbf{x_a}) \\ \text{s.t.} \quad & \boldsymbol{h}(\mathbf{x_a}) + \mathbf{z} = \mathbf{y} + \mathbf{a} = \mathbf{y_a} \\ & \|\mathbf{a}\|_0 \leq c, \end{aligned} \tag{2.35}$$

where $\mathbf{x_a}$ is the corrupted state, $\mathbf{a}$ is the attack vector, $\mathbf{y_a}$ is the attacked observation, and $g(\cdot)$ is the objective of the attacker. The condition on the l_0-norm of the attacked vector allows the attacker to bypass the bad data detector by attacking only a subset of sensors.

Notice that the attacker objective can be different based on its specific goal, for example:

- Target state attack, where $g(\mathbf{x_a}) = \|\mathbf{x_a} - \mathbf{x}_{\text{target}}\|_2^2$, which will put the corrupted state to the targeted value $\mathbf{x}_{\text{target}}$.

- Voltage collapse attack, where $g(\mathbf{x_a}) = \|\mathbf{x_a}\|_2^2$, which will lead the estimator to believe that the voltage is low.

- State deviation attack, where $g(\mathbf{x_a}) = -\|\mathbf{x_a} - \hat{\mathbf{x}}\|_2^2$, which will force the corrupted state to be different from the original predicted one.

Three main types of defense strategies have been developed in the literature. The first one is the detection of the FDIAs [KOSUT and collab., 2010; PASQUALETTI and collab., 2011], which can be used to discard the compromised measurements. The second one focuses on protecting the communication channel between the meters and the control center using encryption, authentication and key management [DÁN and SANDBERG, 2010; TEIXEIRA and collab., 2015; WANG and LU, 2013]. Finally, Robust State Estimation (RSE) has been created to develop state estimations that are robust against bad data. CELIK and ABUR [1992]; KOTIUGA and VIDYASAGAR [1982]; MILI and collab. [1994]; ZHU and GIANNAKIS [2012] focused on the DC model while ZHANG and collab. [2017] investigated a AC-based solution that requires many relaxations.

2.6 The Fisher-Rao measure, and the data-depths

In the following, we will mathematically define the two objects we used during this Ph.D.

2.6.1 The Fisher-Rao distance (FRD)

Consider the family of probability distributions over the classes $\mathscr{C} \doteq \{q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) : \mathbf{x} \in \mathscr{X}\}$ parametrized by $\mathbf{x}$. Assume that the following regularity conditions hold [ATKINSON and MITCHELL, 1981]:

(i) $\nabla_{\mathbf{x}}\, q_{\boldsymbol{\theta}}(y|\mathbf{x})$ exists for all $\mathbf{x}$, y and $\boldsymbol{\theta} \in \Theta$;

(ii) $\sum_{y\in\mathscr{Y}} \nabla_{\mathbf{x}}\, q_{\boldsymbol{\theta}}(y|\mathbf{x}) = 0$ for all $\mathbf{x}$ and $\boldsymbol{\theta} \in \Theta$;

(iii) $\mathbf{G}(\mathbf{x}) = \mathbb{E}_{\mathrm{Y}\sim q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})}\left[\nabla_{\mathbf{x}} \log q_{\boldsymbol{\theta}}(\mathrm{Y}|\mathbf{x}) \nabla_{\mathbf{x}}^{\mathsf{T}} \log q_{\boldsymbol{\theta}}(\mathrm{Y}|\mathbf{x})\right]$ is positive definite for any $\mathbf{x}$ and $\boldsymbol{\theta} \in \Theta$.

The variance of the differential form $\nabla_{\mathbf{x}}^{\mathsf{T}} \log q_{\boldsymbol{\theta}}(\mathrm{Y}|\mathbf{x})\, d\mathbf{x}$ can then be interpreted as the square of a differential arc length ds^2 in the space $\mathscr{C}$, which yields

$$ds^2 = \langle d\mathbf{x}, d\mathbf{x}\rangle_{\mathbf{G}(\mathbf{x})} = d\mathbf{x}^{\mathsf{T}} \mathbf{G}(\mathbf{x})\, d\mathbf{x}. \tag{2.36}$$

Thus, $\mathbf{G}$, which is the Fisher Information Matrix (FIM), can be adopted as a metric tensor. We now consider a curve $\boldsymbol{\gamma} : [0,1] \to \mathscr{X}$ in the input space connecting two arbitrary points $\mathbf{x}$ and $\mathbf{x}'$, i.e., such that $\boldsymbol{\gamma}(0) = \mathbf{x}$ and $\boldsymbol{\gamma}(1) = \mathbf{x}'$. Notice that this curve induces the following curve in the space $\mathscr{C}$: $q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{\gamma}(t))$ for $t \in [0,1]$. The Fisher-Rao distance between the distributions $q_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ and $q'_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')$ will be denoted as $d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}})$ and is formally defined as:

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) \doteq \inf_{\boldsymbol{\gamma}} \int_0^1 \sqrt{\frac{d\boldsymbol{\gamma}^{\mathsf{T}}(t)}{dt} \mathbf{G}(\boldsymbol{\gamma}(t)) \frac{d\boldsymbol{\gamma}(t)}{dt}}, \tag{2.37}$$

where the infimum is taken over all piecewise smooth curves. This means that the FRD is the length of the *geodesic* between points $\mathbf{x}$ and $\mathbf{x}'$ using the FIM as the metric tensor. Several examples for simple families of distributions can be found in ATKINSON and MITCHELL [1981].

2.6.2 The data-depths

A data depth function, formally defined as:

$$\begin{aligned} \mathrm{D}: \quad \mathbb{R}^d \times \mathscr{P}(\mathbb{R}^d) &\longrightarrow [0,1], \\ (\mathbf{x}, \mathrm{P}) &\longmapsto \mathrm{D}(\mathbf{x}, \mathrm{P}), \end{aligned} \tag{2.38}$$

where $\mathscr{P}(\mathbb{R}^d)$ denotes the space of all probability distributions on $\mathbb{R}^d$, measures the centrality of any element in $\mathbf{x} \in \mathbb{R}^d$ w.r.t. a probability distribution P (respectively, a data set). It provides a center-outward ordering of points in the support of P and can be straightforwardly used to extend the notions of rank or order statistics to multivariate data. The higher $D(\mathbf{x}, P)$, the deeper $\mathbf{x}$ is in P.

Since data depth naturally way defines a non-parametric pre-order on $\mathbb{R}^d$ w.r.t. a probability distribution, it can be seen as a centrality-based alternative to the cumulative distribution function for multivariate data. Many different definition of data-depths, respecting Equation 2.38, has been proposed CHEN and collab. [2015]; CUEVAS and collab. [2007]; KOSHEVOY and MOSLER [1997]; LIU [1990]; RAMSAY and collab. [2019]; STAERMAN and collab. [2021]; ZUO [2003], each presenting its own theoretical and practical properties.

However, several axioms have been developed throughout the recent decades a "good" depth function should satisfy DYCKERHOFF [2004]; LIU [1990]; ZUO and SERFLING [2000]:

($\mathbf{D}_1$) (AFFINE INVARIANCE) Denoting by $P_{\mathbf{X}}$ the distribution of any r.v. $\mathbf{X}$ taking its values in $\mathbb{R}^d$, we have:

$$\forall \mathbf{x} \in \mathbb{R}^d,\ D(\mathbf{Ax}+\mathbf{b}, P_{\mathbf{AX}+\mathbf{b}}) = D(\mathbf{x}, P_{\mathbf{X}}),$$

for any $d \times d$ nonsingular matrix $\mathbf{A}$ with real entries and any vector $\mathbf{b}$ in $\mathbb{R}^d$.

($\mathbf{D}_2$) (MAXIMALITY AT CENTER) For any $P \in \mathscr{P}(\mathbb{R}^d)$ that has a symmetry center $\mathbf{x}^*$ (in a sense to be specified), the depth function $D(., P)$ takes its maximum value at it:

$$D(\mathbf{x}^*, P) = \sup_{\mathbf{x} \in \mathbb{R}^d} D(\mathbf{x}, P).$$

($\mathbf{D}_3$) (MONOTONICITY RELATIVE TO DEEPEST POINT) For any $P \in \mathscr{P}(\mathbb{R}^d)$ with deepest point $\mathbf{x}^*$, the depth at any point $\mathbf{x}$ in $\mathbb{R}^d$ decreases as one moves away from $\mathbf{x}^*$ along any ray passing through it:

$$\forall \xi \in [0,1],\ D(\mathbf{x}^*, P) \geq D(\mathbf{x}^* + \xi(\mathbf{x} - \mathbf{x}_P), P).$$

($\mathbf{D}_4$) (VANISHING AT INFINITY) For any $P \in \mathscr{P}(\mathbb{R}^d)$, the depth function D vanishes at infinity:

$$D(\mathbf{x}, P) \to 0, \text{ as } ||\mathbf{x}|| \to \infty.$$

Chapter 2 Conclusion

This chapter first presented an overview of the Deep Learning background. Then, we introduced the current state-of-the-art methods to attack neural networks. Later, we reviewed the existing methods to increase the models' robustness against adversarial threats. This will be useful to appreciate better the contribution of Part I. Then, we presented existing detection mechanisms we will compare against in Part II. Then, we presented quickly the state estimation problem for Smart Grids. Finally, we provided a quick definition of the two main tools we used during this thesis. Now that we have presented what exists in the literature, we will present our first contribution on leveraging the knowledge about the models' output space to increase their robustness.

2.7 References

ABUR, A. and A. G. EXPOSITO. 2004, *Power system state estimation: theory and implementation*, CRC press. 68

ALAYRAC, J.-B., J. UESATO, P.-S. HUANG, A. FAWZI, R. STANFORTH and P. KOHLI. 2019, «Are labels required for improving adversarial robustness?», in *Advances in Neural Information Processing Systems*, p. 12 214–12 223. 63, 64

ALDAHDOOH, A., W. HAMIDOUCHE and O. DÉFORGES. 2021a, «Revisiting model's uncertainty and confidences for adversarial example detection», *arXiv preprint arXiv: 2103.05354*. 65

ALDAHDOOH, A., W. HAMIDOUCHE, S. A. FEZZA and O. DÉFORGES. 2021b, «Adversarial example detection for DNN models: A review», *arXiv preprint arXiv:2105.00203*. 64

ANDRIUSHCHENKO, M., F. CROCE, N. FLAMMARION and M. HEIN. 2020, «Square attack: a query-efficient black-box adversarial attack via random search», in *European Conference on Computer Vision*, Springer, p. 484–501. 58

ATHALYE, A., N. CARLINI and D. WAGNER. 2018, «Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples», in *International conference on machine learning*, PMLR, p. 274–283. 61

ATKINSON, C. and A. F. S. MITCHELL. 1981, «Rao's distance measure», *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, vol. 43, nº 3, p. 345–365, ISSN 0581572X. 71

ATZMON, M., N. HAIM, L. YARIV, O. ISRAELOV, H. MARON and Y. LIPMAN. 2019, «Controlling neural level sets», in *Advances in Neural Information Processing Systems*, p. 2034–2043. 64

BOURLARD, H. and Y. KAMP. 1988, «Auto-association by multilayer perceptrons and singular value decomposition», *Biological cybernetics*, vol. 59, nº 4, p. 291–294. 52

BUNEL, R. R., I. TURKASLAN, P. TORR, P. KOHLI and P. K. MUDIGONDA. 2018, «A unified view of piecewise linear neural network verification», *Advances in Neural Information Processing Systems*, vol. 31. 61

CAO, X. and N. Z. GONG. 2017, «Mitigating evasion attacks to deep neural networks via region-based classification», in *Proceedings of the 33rd Annual Computer Security Applications Conference*, p. 278–287. 61

CARLINI, N. and D. WAGNER. 2017, «Towards evaluating the robustness of neural networks», in *2017 IEEE Symposium on Security and Privacy (SP)*, ISSN 2375-1207, p. 39–57, doi: 10.1109/SP.2017.49. 57

CARMON, Y., A. RAGHUNATHAN, L. SCHMIDT, J. C. DUCHI and P. S. LIANG. 2019, «Unlabeled data improves adversarial robustness», in *Advances in Neural Information Processing Systems*, p. 11 192–11 203. 63, 64

CARRARA, F., R. BECARELLI, R. CALDELLI, F. FALCHI and G. AMATO. 2018, «Adversarial examples detection in features distance spaces», in *Computer Vision - ECCV 2018 Workshops - Munich, Germany, Proceedings, Part II*, vol. 11130, Springer, p. 313–327. 65

CELIK, M. K. and A. ABUR. 1992, «A robust wlav state estimator using transformations», *IEEE Transactions on Power Systems*, vol. 7, nº 1, p. 106–113. 70

CHEN, B., K. M. TING, T. WASHIO and G. HAFFARI. 2015, «Half-space mass: a maximally robust and efficient data depth method», *Machine Learning*, vol. 100, nº 2, p. 677–699. 72

CHEN, J., M. I. JORDAN and M. J. WAINWRIGHT. 2020, «Hopskipjumpattack: A query-efficient decision-based attack», in *2020 ieee symposium on security and privacy (sp)*, IEEE, p. 1277–1294. 59

CHENG, Z., H. SUN, M. TAKEUCHI and J. KATTO. 2018, «Deep convolutional autoencoder-based lossy image compression», in *2018 Picture Coding Symposium (PCS)*, IEEE, p. 253–257. 52

CHO, K., B. VAN MERRIËNBOER, C. GULCEHRE, D. BAHDANAU, F. BOUGARES, H. SCHWENK and Y. BENGIO. 2014, «Learning phrase representations using rnn encoder-decoder for statistical machine translation», *arXiv preprint arXiv:1406.1078*. 51

CHOW, C.-K. 1957, «An optimum character recognition system using decision functions», *IRE Transactions on Electronic Computers*, , nº 4, p. 247–254. 64

CHUNG, J., C. GULCEHRE, K. CHO and Y. BENGIO. 2014, «Empirical evaluation of gated recurrent neural networks on sequence modeling», *arXiv preprint arXiv:1412.3555*. 51

CISSE, M., P. BOJANOWSKI, E. GRAVE, Y. DAUPHIN and N. USUNIER. 2017, «Parseval networks: Improving robustness to adversarial examples», in *International Conference on Machine Learning*, PMLR, p. 854–863. 61

COHEN, J., E. ROSENFELD and Z. KOLTER. 2019, «Certified adversarial robustness via randomized smoothing», in *International Conference on Machine Learning*, PMLR, p. 1310–1320. 61

CROCE, F., M. ANDRIUSHCHENKO, V. SEHWAG, E. DEBENEDETTI, N. FLAMMARION, M. CHIANG, P. MITTAL and M. HEIN. 2020, «Robustbench: a standardized adversarial robustness benchmark», *arXiv preprint arXiv:2010.09670*. 60

CROCE, F. and M. HEIN. 2020a, «Minimally distorted adversarial examples with a fast adaptive boundary attack», in *International Conference on Machine Learning*, PMLR, p. 2196–2205. 11, 57

CROCE, F. and M. HEIN. 2020b, «Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks», in *International Conference on Machine Learning*, PMLR, p. 2206–2216. 55, 60

CUEVAS, A., M. FEBRERO and R. FRAIMAN. 2007, «Robust estimation and classification for functional data via projection-based depth notions», *Computational Statistics*, vol. 22, nº 3, p. 481–496. 72

DÁN, G. and H. SANDBERG. 2010, «Stealth attacks and protection schemes for state estimators in power systems», in *2010 first IEEE international conference on smart grid communications*, IEEE, p. 214–219. 69, 70

DEVLIN, J., M.-W. CHANG, K. LEE and K. TOUTANOVA. 2019, «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, Volume 1 (Long and Short*

Papers), Association for Computational Linguistics, Minneapolis, Minnesota, p. 4171–4186, doi: 10.18653/v1/N19-1423. 51

DOSOVITSKIY, A., L. BEYER, A. KOLESNIKOV, D. WEISSENBORN, X. ZHAI, T. UNTERTHINER, M. DEHGHANI, M. MINDERER, G. HEIGOLD, S. GELLY and collab.. 2020, «An image is worth 16x16 words: Transformers for image recognition at scale», *arXiv preprint arXiv:2010.11929*. 11, 51

DYCKERHOFF, R. 2004, «Data depth satisfying the projection property», *Allgemeines Statistisches Archiv*, vol. 88, nº 2, p. 163–190. 72

ENGSTROM, L., B. TRAN, D. TSIPRAS, L. SCHMIDT and A. MADRY. 2019, «Exploring the landscape of spatial robustness», in *International Conference on Machine Learning*, PMLR, p. 1802–1811. 59

FEINMAN, R., R. R. CURTIN, S. SHINTRE and A. B. GARDNER. 2017, «Detecting adversarial samples from artifacts», *arXiv preprint arXiv:1703.00410*. 65, 66

GIANNAKIS, G. B., V. KEKATOS, N. GATSIS, S.-J. KIM, H. ZHU and B. F. WOLLENBERG. 2013, «Monitoring and optimization for power grids: A signal processing perspective», *IEEE Signal Processing Magazine*, vol. 30, nº 5, p. 107–128. 68

GOGNA, A. and A. MAJUMDAR. 2019, «Discriminative autoencoder for feature extraction: Application to character recognition», *Neural Processing Letters*, vol. 49, nº 3, p. 1723–1735. 52

GOODFELLOW, I. J., J. SHLENS and C. SZEGEDY. 2015, «Explaining and harnessing adversarial examples», *International Conference on Learning Representations*. 11, 54, 55, 61, 62

GOUK, H., E. FRANK, B. PFAHRINGER and M. J. CREE. 2021, «Regularisation of neural networks by enforcing lipschitz continuity», *Machine Learning*, vol. 110, nº 2, p. 393–416. 61

GROSSE, K., P. MANOHARAN, N. PAPERNOT, M. BACKES and P. D. MCDANIEL. 2017, «On the (statistical) detection of adversarial examples», *arXiv preprint arXiv:1702.06280*. 65

HAYKIN, S. 1994, *Neural networks: a comprehensive foundation*, Prentice Hall PTR. 49

HE, K., X. ZHANG, S. REN and J. SUN. 2016, «Deep residual learning for image recognition», in *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 770–778. 11, 50

HEIN, M. and M. ANDRIUSHCHENKO. 2017, «Formal guarantees on the robustness of a classifier against adversarial manipulation», *Advances in neural information processing systems*, vol. 30. 61

HEINRICH, B. S. 2018, «Energy atlas 2018. figures and facts about renewables in europe», . 11, 68

HENDRYCKS, D., K. LEE and M. MAZEIKA. 2019, «Using pre-training can improve model robustness and uncertainty», in *International Conference on Machine Learning*, PMLR, p. 2712–2721. 64

HINTON, G., O. VINYALS and J. DEAN. 2015, «Distilling the knowledge in a neural network», in *NIPS Deep Learning and Representation Learning Workshop*. 61

HOCHREITER, S. and J. SCHMIDHUBER. 1997, «Long short-term memory», *Neural computation*, vol. 9, nº 8, p. 1735–1780. 51

HUANG, L., C. ZHANG and H. ZHANG. 2020, «Self-adaptive training: beyond empirical risk minimization», *Advances in Neural Information Processing Systems*, vol. 33. 64

HUG, G. and J. A. GIAMPAPA. 2012, «Vulnerability assessment of ac state estimation with respect to false data injection cyber-attacks», *IEEE Transactions on smart grid*, vol. 3, nº 3, p. 1362–1370. 69

JIN, M., J. LAVAEI and K. H. JOHANSSON. 2018, «Power grid ac-based state estimation: Vulnerability analysis against cyber attacks», *IEEE Transactions on Automatic Control*, vol. 64, nº 5, p. 1784–1799. 69, 70

KEKATOS, V., G. WANG, H. ZHU and G. B. GIANNAKIS. 2017, «Psse redux: Convex relaxation, decentralized, robust, and dynamic approaches», *arXiv preprint arXiv:1708.03981*. 69

KHERCHOUCHE, A., S. A. FEZZA, W. HAMIDOUCHE and O. DÉFORGES. 2020, «Natural scene statistics for detecting adversarial examples in deep neural networks», in *22nd IEEE International Workshop on Multimedia Signal Processing*, IEEE, p. 1–6. 11, 65, 66

KOSHEVOY, G. and K. MOSLER. 1997, «Zonoid trimming for multivariate distributions», *The Annals of Statistics*, vol. 25, nº 5, p. 1998–2017. 72

KOSUT, O., L. JIA, R. J. THOMAS and L. TONG. 2010, «Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures», in *2010 first IEEE international conference on smart grid communications*, IEEE, p. 220–225. 69, 70

KOTIUGA, W. W. and M. VIDYASAGAR. 1982, «Bad data rejection properties of weughted least absolute value techniques applied to static state estimation», *IEEE Transactions on Power Apparatus and Systems*, , nº 4, p. 844–853. 70

KRIZHEVSKY, A., I. SUTSKEVER and G. E. HINTON. 2012, «Imagenet classification with deep convolutional neural networks», *Advances in neural information processing systems*, vol. 25. 50

KURAKIN, A., I. J. GOODFELLOW and S. BENGIO. 2018, «Adversarial examples in the physical world», in *Artificial intelligence safety and security*, Chapman and Hall/CRC, p. 99–112. 55

LECUN, Y., B. BOSER, J. DENKER, D. HENDERSON, R. HOWARD, W. HUBBARD and L. JACKEL. 1989, «Handwritten digit recognition with a back-propagation network», *Advances in neural information processing systems*, vol. 2. 11, 49, 50

LECUYER, M., V. ATLIDAKIS, R. GEAMBASU, D. HSU and S. JANA. 2019, «Certified robustness to adversarial examples with differential privacy», in *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, p. 656–672. 61

LEE, K., K. LEE, H. LEE and J. SHIN. 2018, «A simple unified framework for detecting out-of-distribution samples and adversarial attacks», in *Advances in Neural Information Processing Systems 31*, édité par S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett, Curran Associates, Inc., p. 7167–7177. 65

LI, X. and F. LI. 2017, «Adversarial examples detection in deep networks with convolutional filter statistics», in *IEEE International Conference on Computer Vision, ICCV*, IEEE Computer Society, p. 5775–5783. 65

LIANG, B., H. LI, M. SU, X. LI, W. SHI and X. WANG. 2021, «Detecting adversarial image examples in deep neural networks with adaptive noise reduction», *IEEE Trans. Dependable Secur. Comput.*, vol. 18, nº 1, p. 72–85. 65

LIANG, G., J. ZHAO, F. LUO, S. R. WELLER and Z. Y. DONG. 2016, «A review of false data injection attacks against modern power systems», *IEEE Transactions on Smart Grid*, vol. 8, nº 4, p. 1630–1638. 69

LIANG, J., O. KOSUT and L. SANKAR. 2014, «Cyber attacks on ac state estimation: Unobservability and physical consequences», in *2014 IEEE PES General Meeting| Conference & Exposition*, IEEE, p. 1–5. 69

LIN, Z., M. FENG, C. N. D. SANTOS, M. YU, B. XIANG, B. ZHOU and Y. BENGIO. 2017, «A structured self-attentive sentence embedding», *arXiv preprint arXiv:1703.03130*. 51

LIU, R. 1990, «On a notion of data depth based on random simplices», *The Annals of Statistics*, vol. 18, p. 405–414. 72

LIU, X., M. CHENG, H. ZHANG and C.-J. HSIEH. 2018a, «Towards robust neural networks via random self-ensemble», in *Proceedings of the European Conference on Computer Vision (ECCV)*, p. 369–385. 61

LIU, X., Y. LI, C. WU and C.-J. HSIEH. 2018b, «Adv-bnn: Improved adversarial defense through robust bayesian neural network», *arXiv preprint arXiv:1810.01279*. 61

LIU, Y., P. NING and M. K. REITER. 2011, «False data injection attacks against state estimation in electric power grids», *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, nº 1, p. 1–33. 68, 69

LOMUSCIO, A. and L. MAGANTI. 2017, «An approach to reachability analysis for feed-forward relu neural networks», *arXiv preprint arXiv:1706.07351*. 61

LU, J., T. ISSARANON and D. A. FORSYTH. 2017, «Safetynet: Detecting and rejecting adversarial examples robustly», in *IEEE International Conference on Computer Vision*, IEEE Computer Society, p. 446–454. 65

MA, S., Y. LIU, G. TAO, W. LEE and X. ZHANG. 2019, «NIC: detecting adversarial samples with neural network invariant checking», in *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*, The Internet Society. 65

MA, X., B. LI, Y. WANG, S. M. ERFANI, S. N. R. WIJEWICKREMA, G. SCHOENEBECK, D. SONG, M. E. HOULE and J. BAILEY. 2018, «Characterizing adversarial subspaces using local intrinsic dimensionality», in *6th International Conference on Learning Representations*. 65

MADRY, A., A. MAKELOV, L. SCHMIDT, D. TSIPRAS and A. VLADU. 2018, «Towards deep learning models resistant to adversarial attacks», in *International Conference on Learning Representations*. 54, 55, 62

MENG, D. and H. CHEN. 2017, «Magnet: A two-pronged defense against adversarial examples», in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, édité par B. M. Thuraisingham, D. Evans, T. Malkin and D. Xu, ACM, p. 135–147. 65, 67

METZEN, J. H., T. GENEWEIN, V. FISCHER and B. BISCHOFF. 2017, «On detecting adversarial perturbations», in *5th International Conference on Learning Representations*. 65

MILI, L., M. G. CHENIAE and P. J. ROUSSEEUW. 1994, «Robust state estimation of electric power systems», *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 41, nº 5, p. 349–358. 70

MOOSAVI-DEZFOOLI, S., A. FAWZI and P. FROSSARD. 2016, «Deepfool: A simple and accurate method to fool deep neural networks», in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ISSN 1063-6919, p. 2574–2582, doi: 10.1109/CVPR.2016.282. 11, 56

NAIR, V. and G. E. HINTON. 2010, «Rectified linear units improve restricted boltzmann machines», in *Icml*. 49

PANG, T., C. DU, Y. DONG and J. ZHU. 2018, «Towards robust detection of adversarial examples», in *Advances in Neural Information Processing Systems 31*, p. 4584–4594. 65

PAPERNOT, N., P. MCDANIEL, I. GOODFELLOW, S. JHA, Z. B. CELIK and A. SWAMI. 2017, «Practical black-box attacks against machine learning», in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, p. 506–519. 58

PAPERNOT, N., P. D. MCDANIEL and I. J. GOODFELLOW. 2016a, «Transferability in machine learning: from phenomena to black-box attacks using adversarial samples», *CoRR*, vol. abs/1605.07277. 61

PAPERNOT, N., P. D. MCDANIEL, S. JHA, M. FREDRIKSON, Z. B. CELIK and A. SWAMI. 2016b, «The limitations of deep learning in adversarial settings», *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, p. 372–387. 58

PASQUALETTI, F., R. CARLI and F. BULLO. 2011, «A distributed method for state estimation and false data detection in power networks», in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, p. 469–474. 69, 70

RAGHUNATHAN, A., J. STEINHARDT and P. LIANG. 2018a, «Certified defenses against adversarial examples», in *International Conference on Learning Representations*. 61

RAGHUNATHAN, A., J. STEINHARDT and P. S. LIANG. 2018b, «Semidefinite relaxations for certifying robustness to adversarial examples», *Advances in Neural Information Processing Systems*, vol. 31. 61

RAMSAY, K., S. DUROCHER and A. LEBLANC. 2019, «Integrated rank-weighted depth», *Journal of Multivariate Analysis*, vol. 173, p. 51–69. 72

REZENDE, D. J., S. MOHAMED and D. WIERSTRA. 2014, «Stochastic backpropagation and approximate inference in deep generative models», in *International conference on machine learning*, PMLR, p. 1278–1286. 52

RICE, L., E. WONG and Z. KOLTER. 2020, «Overfitting in adversarially robust deep learning», in *International Conference on Machine Learning*, PMLR, p. 8093–8104. 64

SANDBERG, H., A. TEIXEIRA and K. H. JOHANSSON. 2010, «On security indices for state estimators in power networks», in *First Workshop on Secure Control Systems (SCS), Stockholm, 2010*. 69

SOLTAN, S., M. YANNAKAKIS and G. ZUSSMAN. 2016, «Power grid state estimation following a joint cyber and physical attack», *IEEE Transactions on Control of Network Systems*, vol. 5, nº 1, p. 499–512. 70

SOLTAN, S., M. YANNAKAKIS and G. ZUSSMAN. 2018, «React to cyber attacks on power grids», *IEEE Transactions on Network Science and Engineering*, vol. 6, nº 3, p. 459–473. 70

SOLTAN, S. and G. ZUSSMAN. 2018, «Expose the line failures following a cyber-physical attack on the power grid», *IEEE Transactions on Control of Network Systems*, vol. 6, nº 1, p. 451–461. 70

SOTGIU, A., A. DEMONTIS, M. MELIS, B. BIGGIO, G. FUMERA, X. FENG and F. ROLI. 2020, «Deep neural rejection against adversarial examples», *EURASIP J. Inf. Secur.*, vol. 2020, p. 5. 65

STAERMAN, G., P. MOZHAROVSKYI and S. CLÉMENÇON. 2021, «Affine-invariant integrated rank-weighted depth: Definition, properties and finite sample analysis», *arXiv preprint arXiv:2106.11068*. 72

SZEGEDY, C., W. ZAREMBA, I. SUTSKEVER, J. BRUNA, D. ERHAN, I. GOODFELLOW and R. FERGUS. 2014a, «Intriguing properties of neural networks», *International Conference on Learning Representations*. 53, 54

SZEGEDY, C., W. ZAREMBA, I. SUTSKEVER, J. BRUNA, D. ERHAN, I. J. GOODFELLOW and R. FERGUS. 2014b, «Intriguing properties of neural networks», in *2nd International Conference on Learning Representations*. 55

TEIXEIRA, A., K. C. SOU, H. SANDBERG and K. H. JOHANSSON. 2015, «Secure control systems: A quantitative risk management approach», *IEEE Control Systems Magazine*, vol. 35, nº 1, p. 24–45. 69, 70

THEIS, L., W. SHI, A. CUNNINGHAM and F. HUSZÁR. 2017, «Lossy image compression with compressive autoencoders», *arXiv preprint arXiv:1703.00395*. 52

TRAMÈR, F., N. PAPERNOT, I. GOODFELLOW, D. BONEH and P. MCDANIEL. 2017, «The space of transferable adversarial examples», *arXiv preprint arXiv:1704.03453*. 62

VASWANI, A., N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER and I. POLOSUKHIN. 2017, «Attention is all you need», *Advances in neural information processing systems*, vol. 30. 51

VINCENT, P., H. LAROCHELLE, Y. BENGIO and P.-A. MANZAGOL. 2008, «Extracting and composing robust features with denoising autoencoders», in *Proceedings of the 25th international conference on Machine learning*, p. 1096–1103. 52

WANG, W. and Z. LU. 2013, «Cyber security in the smart grid: Survey and challenges», *Computer networks*, vol. 57, nº 5, p. 1344–1371. 70

WANG, Y., D. ZOU, J. YI, J. BAILEY, X. MA and Q. GU. 2019, «Improving adversarial robustness requires revisiting misclassified examples», in *International Conference on Learning Representations*. 63

WONG, E. and J. Z. KOLTER. 2020, «Learning perturbation sets for robust machine learning», *arXiv preprint arXiv:2007.08450*. 61

WONG, E., F. SCHMIDT, J. H. METZEN and J. Z. KOLTER. 2018, «Scaling provable adversarial defenses», in *Advances in Neural Information Processing Systems*, p. 8400–8409. 61

WU, D., S.-T. XIA and Y. WANG. 2020, «Adversarial weight perturbation helps robust generalization», *Advances in Neural Information Processing Systems*, vol. 33. 64

XIE, C., J. WANG, Z. ZHANG, Z. REN and A. YUILLE. 2017, «Mitigating adversarial effects through randomization», *arXiv preprint arXiv:1711.01991*. 61

XIE, L., Y. MO and B. SINOPOLI. 2010, «False data injection attacks in electricity markets», in *2010 First IEEE International Conference on Smart Grid Communications*, IEEE, p. 226–231. 69

XU, W., D. EVANS and Y. QI. 2018, «Feature squeezing: Detecting adversarial examples in deep neural networks», in *25th Annual Network and Distributed System Security Symposium*, The Internet Society. 65, 67

YASENKO, L., Y. KLYATCHENKO and O. TARASENKO-KLYATCHENKO. 2020, «Image noise reduction by denoising autoencoder», in *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, IEEE, p. 351–355. 52

YUAN, Y., Z. LI and K. REN. 2011, «Modeling load redistribution attacks in power systems», *IEEE Transactions on Smart Grid*, vol. 2, nº 2, p. 382–390. 69

ZHANG, H., Y. YU, J. JIAO, E. P. XING, L. E. GHAOUI and M. I. JORDAN. 2019, «Theoretically principled trade-off between robustness and accuracy», in *International Conference on Machine Learning*, p. 1–11. 62

ZHANG, Y., R. MADANI and J. LAVAEI. 2017, «Conic relaxations for power system state estimation with line measurements», *IEEE Transactions on Control of Network Systems*, vol. 5, nº 3, p. 1193–1205. 70

ZHENG, Z. and P. HONG. 2018, «Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks», in *Advances in Neural Information Processing Systems 31*, p. 7924–7933. 65

ZHU, H. and G. B. GIANNAKIS. 2012, «Robust power system state estimation for the nonlinear ac flow model», in *2012 North American Power Symposium (NAPS)*, IEEE, p. 1–6. 69, 70

ZUO, Y. 2003, «Projection-based depth functions and associated medians», *The Annals of Statistics*, vol. 31, nº 5, p. 1460–1490. 72

ZUO, Y. and R. SERFLING. 2000, «General notions of statistical depth function», *The Annals of Statistics*, vol. 28, nº 2, p. 461–482. 72

Part I

PART 1: Information-Geometric Methods for Adversarial Robustness and its Applications to Smart Grids

Part I Abstract

This part is dedicated to our proposed answer to the first question asked in Chapter 1: **How can we use the internal structure of DNN's output space to improve its robustness ?** This part is split into three chapters consisting of three different contributions.

- In Chapter 3, we provide our solution to improve adversarial robustness through the use of a regularizer based on an information-geometric measure: the Fisher-Rao measure. The output space of a neural network forms a statistical manifold, and the decision is taken on it. To leverage this knowledge, we decided to use a measure that captures the distance between two outputs following the shape of the statistical manifold at hand: the Fisher-Rao distance. We derived a close-form of the Fisher-Rao distance in the case of a binary classification as well as for the multi-class scenario. On a toy example, we show that it is possible to achieve Pareto-optimum points unachievable using other regularizers. The experimental results show that our regularizer consistently enhances the robustness of neural networks.

- In Chapter 4, we apply our method to the state estimation study in Smart Grids applications. Due to monitoring and service needs, Smart Grid systems include a cyber component that makes them vulnerable to potential threats. In this work, we made use of a Variational AutoEncoder and the physical knowledge of the grid to train an unsupervised state estimator. We design a defensive scheme similar to a min-max problem to craft a robust state estimator using a similar method as the one used in Chapter 3. We experimentally show that it is possible to train robust state estimators for multiple electrical systems in the linearized case.

- In Chapter 4, we assume that the system in linear. Since this assumption might not be realistic for all the bus systems, we propose an extension of the previously introduced framework in Chapter 5. This extension to the non-linear case (i.e., under the AC model assumption) of the state estimation problem is a more general scenario given that all power systems are intrinsically non-linear.

Chapter 3

Adversarial Robustness via Fisher-Rao Regularization

Chapter 3 Abstract

This chapter addresses the first research question dedicated to increasing adversarial robustness. We present our first contribution to the scope of image data. One way to ensure that an image classifier remains trustworthy under threats is to force the model to classify similarly clean and attacked inputs. To do so, a successful line of work is to use a modified training procedure that will take into account perturbed samples via a custom risk. The output of a neural network forms a statistical manifold, and the decision mechanisms depend on this manifold. To leverage this knowledge, we decided to use an information-geometric distance, i.e., the Fisher-Rao distance, that measures the distance between two possible outputs and to use it to train a robust classifier. In this work, we derived the closed form of the Fisher-Rao distance on both deep binary classifiers and multi-class deep classifiers. We later used this formula of the Fisher-Rao distance to define a regularizer that, when minimized, will force the resulting deep classifier to estimate the natural and attacked input in a similar manner. We experimentally showed that using this regularizer will allow the resulting model to achieve Pareto-optimum points impossible to reach when using SOTA regularizers. We also experimentally showed that using Fisher-Rao as a robust regularizer allowed us to increase the robustness of neural networks.

Contents

Abstract

Adversarial robustness has become a topic of growing interest in machine learning since it was observed that neural networks tend to be brittle. We propose an information-geometric formulation of adversarial defense and introduce FIRE, a new Fisher-Rao regularization for the categorical cross-entropy loss, which is based on the geodesic distance between the softmax outputs corresponding to natural and perturbed input features. Based on the information-geometric properties of the class of softmax distributions, we derive an explicit characterization of the Fisher-Rao Distance (FRD) for the binary and multiclass cases, and draw some interesting properties as well as connections with standard regularization metrics. Furthermore, we verify on a simple linear and Gaussian model, that all Pareto-optimal points in the accuracy-robustness region can be reached by FIRE while other state-of-the-art methods fail. Empirically, we evaluate the performance of various classifiers trained with the proposed loss on standard datasets, showing up to a simultaneous 1% of improvement

in terms of clean and robust performances while reducing the training time by 20% over the best-performing methods.

3.1 Introduction

Deep Neural Networks (DNNs) have achieved several breakthroughs in different fields such as computer vision, speech recognition, and Natural Language Processing (NLP). Nevertheless, it is well-known that these systems are extremely sensitive to small perturbations on the inputs [SZEGEDY and collab., 2014], known as adversarial examples. Formally, an adversarial example represents a corrupted input, characterized by a bounded optimal perturbation from the original vector, designed to fool a specified neural networks' task. Adversarial examples have already proven threatful in several domains, including vision and NLP [ALZANTOT and collab.], hence leading to the emergence of the rich area of adversarial machine learning [VOROBEYCHIK and collab., 2018]. The effectiveness of adversarial examples has been attributed to the linear regime of DNNs [GOODFELLOW and collab., 2015] and the data manifold geometrical structure itself [GILMER and collab., 2018], among other hypotheses. More recently, it has been related to the existence of valuable features for classification but meaningless for humans [ILYAS and collab., 2019].

In this paper, we focus on the so-called white-box attacks, for which the attacker has full access to the model. However, it should be noted that black-box attacks, in which the attacker can only query predictions from the model without access to further information, are also feasible [PAPERNOT and collab., 2017]. The literature on adversarial machine learning is extensive and can be divided into three overlapping groups, studying the generation, detection, and defense aspects. The simplest method to generate adversarial examples is the Fast Gradient Sign Method (FGSM) [GOODFELLOW and collab., 2015], including its iterative variant called Projected Gradient Descent (PGD) [MADRY and collab., 2018]. Although widely used, PGD has a few issues that can lead to overestimating the robustness of a model. AutoAttack [CROCE and HEIN, 2020] has been recently developed to overcome those problems, enabling an effective way to test and compare the different defensive schemes.

A simple approach to cope with corrupted examples is to detect and discard them before classification. For instance, FEINMAN and collab. [2017], ZHENG and HONG [2018], and GROSSE and collab. [2017] present different methods to detect corrupted inputs. Although these ideas can be useful to ensure robustness to outliers (i.e., inputs with large deviations with respect to clean examples), they do not seem to be satisfactory solutions for mild adversarial perturbations. In addition, adversarial detection can generally be bypassed by sophisticated attack methods [CARLINI and WAGNER, 2017].

Recently, several works focused on improving the robustness of neural networks

by investigating various defense mechanisms. For instance, certified defense mechanisms addressed the need for more guarantees on the task performance beyond standard evaluation metrics [COHEN and collab., 2019; CROCE and collab., 2019; GOWAL and collab., 2019; LECUYER and collab., 2019; LI and collab., 2019; MIRMAN and collab., 2018; WONG and collab., 2018; ZHANG and collab., 2020]. These methods aim at training classifiers whose predictions at any input feature will remain constant within a set of neighborhoods around the original input. However, these algorithms do not achieve state-of-the-art performance yet. Also, some approaches tend to rely on convex relaxations of the original problem [RAGHUNATHAN and collab., 2018; WONG and KOLTER, 2018] since directly solving the adversarial problem is not tractable. Although these solutions are promising, it is still not possible to scale them to high-dimensional datasets. Finally, we could mention distillation, initially introduced in HINTON and collab. [2015], and further studied in PAPERNOT and collab. [2016]. The idea of distillation is to use a large DNN (the teacher) to train a smaller one (the student), which can perform with similar accuracy while utilizing a temperature parameter to reduce sensitivity to input variations. The resulting defense strategy may be efficient for some attacks but can be defeated with the standard Carlini-Wagner attack.

In this work, we will focus on the most popular strategy for enhancing robustness, which is based on adversarial training, i.e., learning with an augmented training set containing adversarial examples [GOODFELLOW and collab., 2015].

3.1.1 Summary of contributions

Our work investigates the problem of optimizing the trade-off between accuracy and robustness and advances state-of-the-art methods in very different ways.

- We derive an explicit characterization of the Fisher-Rao Distance (FRD) based on the information-geometric properties of the soft-predictions of the neural classifier. That leads to closed-form expressions of the FRD for the binary and multiclass cases (Theorem 1 and Theorem 2, respectively). We further relate them to well-known regularization metrics (presented in Proposition 1).

- We propose a new formulation of adversarial defense, called FIsher-rao REgularizer (FIRE). It consists of optimizing a regularized loss, which encourages the predictions of natural and perturbed samples to be close to each other, according to the manifold of the softmax distributions induced by the neural network. Our loss in Equation 3.7 consists of two terms: the categorical cross-entropy, which favors natural accuracy, and a Fisher-Rao regularization term, which increases adversarial robustness. Furthermore, we prove for a simple logistic regression and Gaussian model that all Pareto-optimal points in the accuracy-robustness region

can be reached by FIRE, while state-of-the-art methods fail (cf. Section 3.4 and Proposition 2).

- Experimentally, on standard benchmarks, we found that FIRE provides an improvement up to roughly 2% of robust accuracy compared to the widely used Kullback-Leibler regularizer [ZHANG and collab., 2019]. We also observed significant improvements over other state-of-the-art methods. In addition, our method typically requires, on average, less computation time (measured by the training runtime on the same GPU cluster) than state-of-the-art methods.

3.1.2 Related work

Adversarial training. Adversarial training (AT) [GOODFELLOW and collab., 2015] is one of the few defenses that has not been broken so far. Indeed, different variations of this method have been proposed. It is based on an attack-defense scheme where the attacker's goal is to create perturbated inputs by maximizing a loss to fool the classifier, while the defender's goal is to classify those attacked inputs rightfully.

Inner attack generation. The inner attacker design is vital to AT since it has to create meaningful attacks. One of the most popular algorithms to generate adversarial examples is Projected Gradient Descent (PGD) [MADRY and collab., 2018], which is an iterative attack: the output at each step is the addition of the previous output and the sign of the loss gradient modulated by a fixed step size. The loss maximized in PGD is often the same loss that is minimized for the defense.

Robust defense loss. An essential choice of the defense mechanism is the robust loss used to attack and defend the network. Initially, GOODFELLOW and collab. [2015]; MADRY and collab. [2018] used the adversarial cross-entropy. However, it was shown that one way to improve adversarial training is through the choice of this loss. TRADES [ZHANG and collab., 2019] introduces a robustness regularizer based on the Kullback-Leibler divergence. MART [WANG and collab., 2019] uses a robustness regularizer that considers the misclassified inputs and boosted losses.

Additional improvement. Whether it is AT, TRADES or MART, they all have been improved in recent years. Those improvements can either rely on pretraining [HENDRYCKS and collab., 2019], early stopping [RICE and collab., 2020], curriculum learning [ATZMON and collab., 2019], adaptative models [HUANG and collab., 2020], unlabeled data to improve generalization [ALAYRAC and collab., 2019; CARMON and collab., 2019] or additional perturbations on the model weights [WU and collab., 2020].

It should be noted that the main disadvantage of adversarial training-based methods remains the required computational expenses. Nevertheless, as will be shown in Section 3.5, FIRE can significantly reduce them.

3.2 Background

We consider a standard supervised learning framework where $\mathbf{x} \in \mathscr{X} \subseteq \mathbb{R}^n$ denotes the input vector on the space $\mathscr{X}$ and $y \in \mathscr{Y}$ the class variable, where $\mathscr{Y} := \{1, \dots, \mathrm{M}\}$. The unknown data distribution is denoted by $p(\mathbf{x}, y) = p(\mathbf{x})\, p(y|\mathbf{x})$. We define a classifier to be a parametric soft-probability model of $p(y|\mathbf{x})$, denoted as $q_{\boldsymbol{\theta}}(y|\mathbf{x})$, where $\boldsymbol{\theta} \in \Theta$ are the parameters. This can be readily used to induce a hard decision: $f_{\boldsymbol{\theta}} : \mathscr{X} \to \mathscr{Y}$ with $f_{\boldsymbol{\theta}}(\mathbf{x}) := \arg\max_{y \in \mathscr{Y}} q_{\boldsymbol{\theta}}(y|\mathbf{x})$. Adversarial examples are denoted as $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta} \in \mathscr{X}$, where $\|\boldsymbol{\delta}\| \leq \varepsilon$ for an arbitrary norm $\|\cdot\|$. Loss functions are denoted as $\ell(\mathbf{x}, \mathbf{x}', y, \boldsymbol{\theta})$ and the corresponding risk functions by $\mathrm{L}(\boldsymbol{\theta})$. We also define the natural missclassification probability as $\mathrm{P}_e(\boldsymbol{\theta}) \doteq \mathbb{P}(f_{\boldsymbol{\theta}}(\mathbf{X}) \neq \mathrm{Y})$, the adversarial missclassification probability as $\mathrm{P}'_e(\boldsymbol{\theta}) \doteq \mathbb{P}(f_{\boldsymbol{\theta}}(\mathbf{X}') \neq \mathrm{Y})$.

3.2.1 Adversarial learning

We provide some background on adversarial learning, focusing on adversarial defense's most popular proposed loss functions. Adversarial examples are slightly modified inputs that can fool a target classifier. Concretely, SZEGEDY and collab. [2014] define the adversarial generation problem as:

$$\mathbf{x}' = \arg\min_{\mathbf{x}' : \|\mathbf{x}' - \mathbf{x}\| \leq \varepsilon} \|\mathbf{x}' - \mathbf{x}\| \ \text{ s.t. } \ f_{\boldsymbol{\theta}}(\mathbf{x}') \neq y, \tag{3.1}$$

where y is the true label (supervision) associated to the sample $\mathbf{x}$. This formulation shows that the vulnerable points of a classifier are the ones close to its decision boundaries. Since this problem is difficult to tackle, it is commonly relaxed as follows [MADRY and collab., 2018]:

$$\mathbf{x}' = \arg\max_{\mathbf{x}' : \|\mathbf{x}' - \mathbf{x}\| \leq \varepsilon} \ell(\mathbf{x}, \mathbf{x}', y, \boldsymbol{\theta}). \tag{3.2}$$

Once adversarial examples are obtained, they can be used to learn a robust classifier as discussed next.

The adversarial problem has been presented in MADRY and collab. [2018] as follows:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{p(\mathbf{x}, y)} \left[\max_{\mathbf{x}' : \|\mathbf{x}' - \mathbf{x}\|_p \leq \varepsilon} \ell(\mathbf{x}, \mathbf{x}', y, \boldsymbol{\theta}) \right], \tag{3.3}$$

where ε denotes the maximal distortion allowed in the adversarial examples according to the l_p-norm. Since the exact solution to the above inner max problem is generally intractable, a relaxation is proposed by generating an adversarial example using an iterative algorithm such as PGD.

Adversarial Cross-Entropy (ACE)

If we take the loss to be the Cross-Entropy (CE), i.e., $\ell(\mathbf{x}, \mathbf{x}', y, \boldsymbol{\theta}) = -\log q_{\boldsymbol{\theta}}(y|\mathbf{x}')$, we obtain the ACE risk:

$$\mathrm{L}_{\mathrm{ACE}}(\boldsymbol{\theta}) \doteq \mathbb{E}_{p(\mathbf{x},y)}\left[\max_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|\leq\varepsilon} -\log q_{\boldsymbol{\theta}}(y|\mathbf{x}')\right]. \tag{3.4}$$

TRADES

Later ZHANG and collab. [2019] defined a new risk based on a trade-off between natural and adversarial performances, controlled through an hyperparameter λ. The resulting risk is the addition of the natural cross-entropy and the Kullback-Leibler (KL) divergence between natural and adversarial probability distributions:

$$\mathrm{L}_{\mathrm{TRADES}}(\boldsymbol{\theta}) \doteq \mathbb{E}_{p(\mathbf{x},y)}\left[\max_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|\leq\varepsilon} -\log q_{\boldsymbol{\theta}}(y|\mathbf{x}) + \lambda\, \mathrm{KL}(q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})\|\, q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}'))\right], \tag{3.5}$$

where

$$\mathrm{KL}(q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})\|\, q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')) \doteq \mathbb{E}_{q_{\boldsymbol{\theta}}(y|\mathbf{x})}\left[\log \frac{q_{\boldsymbol{\theta}}(y|\mathbf{x})}{q_{\boldsymbol{\theta}}(y|\mathbf{x}')}\right]. \tag{3.6}$$

3.3 Adversarial Robustness with Fisher-Rao Regularization

3.3.1 Information geometry and statistical manifold

Statistics on manifolds and information geometry are two different ways in which differential geometry meets statistics. A statistical manifold can be defined as a parameterized family of probability distributions (or density functions) of interest. It is worth to mention that the concept of statistics on manifolds is very different from manifold learning which is a branch of machine learning where the goal is to learn a latent manifold from valued data. In this paper, we are interested in the statistical manifold obtained when fixing the parameters $\boldsymbol{\theta}$ of a DNN and changing its feature input. We consider the following statistical manifold: $\mathscr{C} \doteq \{q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) : \mathbf{x} \in \mathscr{X}\}$. In particular, the focus is on changes in a neighborhood of a particular sample in an adversarial manner (i.e., considering a worst-case perturbation). Please notice that the statistical manifold is different from the loss landscape. The loss landscape is defined as the changes of the risk function with respect to changes in the model parameters (i.e., $\mathrm{L}(\boldsymbol{\theta})$ vs $\boldsymbol{\theta}$), while the statistical manifold refers to the changes of the soft-probabilities of the classifier with respect to changes in the input (i.e., $q_{\boldsymbol{\theta}}(y|\mathbf{x})$ vs $\mathbf{x}$). In order to understand the effect of a perturbation on the input, we first need to be able to capture the distance over the statistical manifold between different probability distributions, i.e., between two different feature inputs. That is precisely

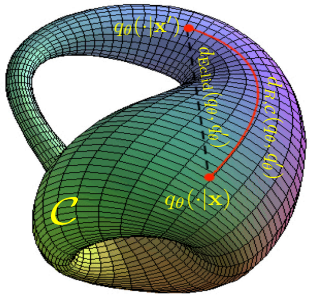
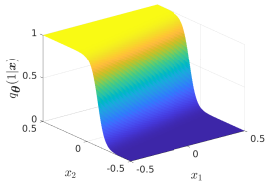


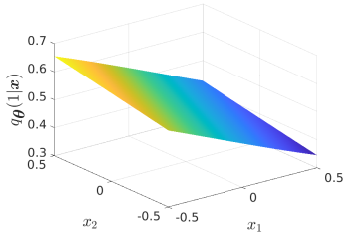
Figure 3.1: Illustration of FRD between two distributions $q_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ and $q'_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')$ over the statistical manifold $\mathscr{C}$. ©2022 IEEE.

what the FRD computes, as illustrated by the red curve in Figure 3.1. It is worth to mention that FRD can be very much different from the euclidean distance since the later does not depend on the shape of the manifold. For a formal mathematical definition of FRD and a short review of basic concepts in information geometry, we refer the reader to Subsection 3.6.1.

As discussed in Section 3.2, the robustness of a classifier is related to the distance between natural examples and the decision boundaries (i.e., points $\mathbf{x}$ such that $q_{\boldsymbol{\theta}}(y|\mathbf{x}) \approx q_{\boldsymbol{\theta}}(y'|\mathbf{x})$ for $y \neq y'$). In fact, if a natural example is far from the decision boundaries, a norm-constrained attack will clearly fail (in this case, the optimization problem in Equation 3.1 will be infeasible). Since the decision boundaries are given by the soft-probabilities $q_{\boldsymbol{\theta}}(y|\mathbf{x})$, this can be equivalently studied by analyzing the shape of the statistical manifold $\mathscr{C}$ (which should not be confused with the loss landscape). In fact, if $q_{\boldsymbol{\theta}}(y|\mathbf{x})$ is relatively flat (i.e., does not change much) with respect to perturbations of $\mathbf{x}$ around $\mathbf{x}_0$, it is clear that adversarial perturbations will not modify the classifier decision at this point. In contrast, if $q_{\boldsymbol{\theta}}(y|\mathbf{x})$ changes sharply with perturbations of $\mathbf{x}$ around $\mathbf{x}_0$, an adversarial can easily leverage this vulnerability to fool the classifier. This notion of robustness is related to the Lipschitz constant of the network, as discussed in various works (e.g., Cisse and collab. [2017]). To illustrate these ideas clearly, let us consider the logistic regression model $q_{\boldsymbol{\theta}}(y|\mathbf{x}) = 1/[1+\exp(-y\boldsymbol{\theta}^\mathsf{T}\mathbf{x})]$, where $n = 2$ and $\mathscr{Y} = \{-1, 1\}$, as a simple example. One way to visualize the statistical manifold $\mathscr{C}$ is to plot $q_{\boldsymbol{\theta}}(1|\mathbf{x})$ as a function of $\mathbf{x}$ (since $q_{\boldsymbol{\theta}}(-1|\mathbf{x}) = 1 - q_{\boldsymbol{\theta}}(1|\mathbf{x})$, this completely characterizes the manifold). This is shown in Figure 3.2a for the value of $\boldsymbol{\theta}$ which minimizes the natural missclassification probability P_e under a conditional Gaussian model for the input $\mathbf{x}$ (see Section 3.4 for details). As can be seen, the manifold is quite sharp around a particular region of $\mathscr{X}$. This region corresponds to the neighborhood of the points for which $\boldsymbol{\theta}^\mathsf{T}\mathbf{x} \approx 0$ as $\mathbf{x}$ is perturbed in the direction of $\boldsymbol{\theta}$. Therefore, we can say that this model is clearly non-robust, as its output can be significantly changed by small perturbations on the input. Consider now the same model but with the values of $\boldsymbol{\theta}$ obtained by minimizing

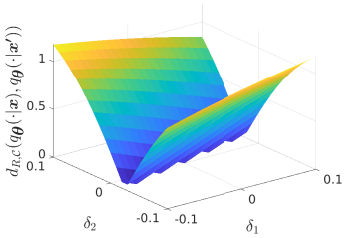


(a) $\boldsymbol{\theta} = [-6.4290, 25.7487]^\intercal$.

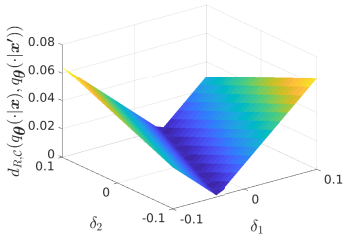


(b) $\boldsymbol{\theta} = [-0.9364, 0.3509]^\intercal$.

Figure 3.2: Visualization of statistical manifold $\mathscr{C}$ defined by the model $q_{\boldsymbol{\theta}}(y|\mathbf{x}) = 1/[1 + \exp(-y\boldsymbol{\theta}^\intercal\mathbf{x})]$ with different values of $\boldsymbol{\theta}$: (a) Parameters minimizing the natural misclassification error probability P_e, (b) Parameters minimizing the adversarial misclassification error probability P'_e. ©2022 IEEE.



(a) $\boldsymbol{\theta} = [-6.4290, 25.7487]^\intercal$.



(b) $\boldsymbol{\theta} = [-0.9364, 0.3509]^\intercal$.

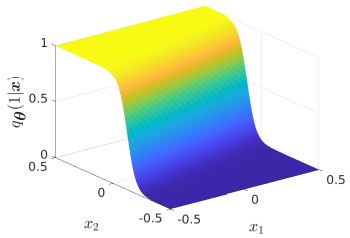
Figure 3.3: FRD between the distributions $q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ and $q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')$ as a function of $\boldsymbol{\delta}$ using the logistic model with different values of $\boldsymbol{\theta}$: (a) Parameters minimizing the natural misclassification error probability P_e, (b) Parameters minimizing the adversarial misclassification error probability P'_e. ©2022 IEEE.

the adversarial missclassification probability P'_e. As can be seen in Figure 3.2b, the statistical manifold is much flatter than in Figure 3.2a, which means that the model is less sensitive to adversarial perturbations on the input. Therefore, it is more robust.

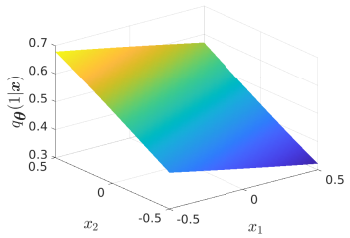
Let us now consider the FRD of the two models around the point $\mathbf{x} = \mathbf{0}$, which gives a point that lies in the decision boundary, by letting $\boldsymbol{\delta} = \mathbf{x}' - \mathbf{x}$ vary in the ℓ_∞ ball $\mathscr{B}_{\infty,\varepsilon} = \{\boldsymbol{\delta} : \|\boldsymbol{\delta}\|_\infty \leq \varepsilon\}$, with $\varepsilon = 0.1$. Figure 3.3a displays the FRD for the parameters $\boldsymbol{\theta}$ which minimize the misclassification error probability P_e, and Figure 3.3b shows the FRD for the parameters $\boldsymbol{\theta}$ which minimize the adversarial misclassification error probability P'_e. Clearly, the abrupt transition of $q_{\boldsymbol{\theta}}(1|\mathbf{x})$ in Figure 3.2a corresponds to a sharp increase on the FRD as $\|\boldsymbol{\delta}\|_\infty$ increases. On the contrary, for a flatter manifold as in Figure 3.2b, the FRD increases much more slowly as $\|\boldsymbol{\delta}\|_\infty$ increases. This example shows how FRD reflects the shape of the statistical manifold $\mathscr{C}$.

Our goal in this work is to use the FRD to control the shape of the statistical manifold by regularizing the misclassification risk.

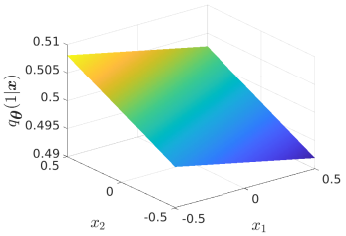
The rest of this section is organized as follows. We begin by introducing the FIRE risk function, which is our main theoretical proposal to improve the robustness of neural networks. We continue with the evaluation of the FRD given by Equation 3.26 for the binary and multiclass classification frameworks and provide some



(a) $\lambda = 0$.



(b) $\lambda = 3.8957$.



(c) $\lambda = 98.1730$.

Figure 3.4: Visualization of statistical manifold $\mathscr{C}$ defined by the model $q_{\boldsymbol{\theta}}(y|\mathbf{x}) = 1/[1 + \exp(-y\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x})]$ when minimizing the FIRE risk function for different values of λ: (a) No adversarial FRD regularization, (b) Medium adversarial FRD regularization, (c) High adversarial FRD regularization. ©2022 IEEE.

exciting properties and connections with other standard distances and well-known information divergences.

3.3.2 The FIRE risk function

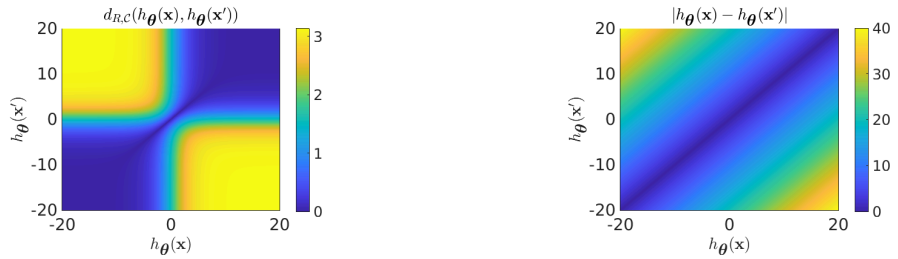
The main proposal of this paper is the FIRE risk function, defined as follows:

$$\mathrm{L}_{\mathrm{FIRE}}(\boldsymbol{\theta}) \doteq \mathbb{E}_{p(\mathbf{x},y)}\left[\max_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|\leq\varepsilon} -\log q_{\boldsymbol{\theta}}(y|\mathbf{x}) + \lambda \cdot d^2_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}), q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}'))\right], \tag{3.7}$$

where $\lambda > 0$ controls the trade-off between natural accuracy and robustness to the adversary.

In Figure 3.4, we show the shape of the statistical manifold $\mathscr{C}$ as λ is varied for the logistic regression model discussed in Subsection 3.3.1 (see also Section 3.4). Notice that when no FRD regularization is used, the manifold in Figure 3.4a is very similar to the one in Figure 3.2a. As the value of λ increases, the weight of the FRD regularization term also increases. As a consequence, the statistical manifold is flattened as expected which is illustrated in Figure 3.4b. However, as shown in Figure 3.4c, setting λ to a very high value causes the statistical manifold to become extremely flat. This means that the model is basically independent of the input, and the classification performance will be poor. Notice the similarities between Figure 3.2 and Figure 3.4.

In what follows, we derive closed-form expressions of the FRD for general classification problems. However, for the sake of clarity, we begin with the binary case.



(a) Fisher-Rao distance.

(b) Euclidean distance.

Figure 3.5: Comparison between FRD and Euclidean distance. ©2022 IEEE.

3.3.3 FRD for the case of binary classification

Let us first consider the binary classification setting, in which $\mathscr{X} \subseteq \mathbb{R}^n$ and $\mathscr{Y} = \{-1, 1\}$ are input and label spaces, respectively. Consider an arbitrary given model:

$$q_{\boldsymbol{\theta}}(y|\mathbf{x}) = \frac{1}{1 + e^{-h_{\boldsymbol{\theta}}(\mathbf{x})y}}. \tag{3.8}$$

Here $h_{\boldsymbol{\theta}}(\mathbf{x})$ represents an arbitrary parametric representation or latent code of the input $\mathbf{x}$. As a matter of fact, we only need to assume that $h_{\boldsymbol{\theta}}$ is a smooth function. The FRD for this model can be computed in closed-form, as shown in the following result. The proof is relegated to Subsection 3.6.2.

Theorem 1 (FRD for binary classifier)**.** *The* FRD *between soft-predictions* $q_{\boldsymbol{\theta}} \equiv q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ *and* $q'_{\boldsymbol{\theta}} \equiv q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')$*, according to Equation 3.8 and corresponding to inputs* $\mathbf{x}$ *and* $\mathbf{x}'$*, is given by*

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) = 2\left|\arctan(e^{h_{\boldsymbol{\theta}}(\mathbf{x}')/2}) - \arctan(e^{h_{\boldsymbol{\theta}}(\mathbf{x})/2})\right|. \tag{3.9}$$

For illustration purposes, Figure 3.5a shows the behavior of the FRD with respect to changes in the latent code compared with the Euclidean distance. It can be observed that the resulting FRD is rather sensitive to variations in the latent space when $h_{\boldsymbol{\theta}}(\mathbf{x}) \approx 0$ while being close to zero for the region in which $|h_{\boldsymbol{\theta}}(\mathbf{x})|$ is large and $|h_{\boldsymbol{\theta}}(\mathbf{x}')| \ll |h_{\boldsymbol{\theta}}(\mathbf{x})|$. This asymmetric behavior is in sharp contrast with the one of the Euclidean distance. However, these quantities are related as shown by the next proposition.

Proposition 1 (FRD vs. Euclidean distance)**.** The Fisher-Rao distance can be bounded as follows:

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) \leq \frac{1}{2}\left|h_{\boldsymbol{\theta}}(\mathbf{x}') - h_{\boldsymbol{\theta}}(\mathbf{x})\right|, \tag{3.10}$$

for any pair of inputs $\mathbf{x}, \mathbf{x}' \in \mathscr{X}$.

The proof of this proposition is relegated to Subsection A.3.1.

Logistic regression:

A particular case of significant importance is that of logistic regression: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^{\mathsf{T}}\mathbf{x}$. In this case, the FRD reduces to:

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) = 2\left|\arctan(e^{\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x}'/2}) - \arctan(e^{\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x}/2})\right|. \tag{3.11}$$

A first-order Taylor approximation in the variable $\boldsymbol{\delta} = \mathbf{x}' - \mathbf{x}$ and maximization over $\boldsymbol{\delta}$ such that $\|\boldsymbol{\delta}\| \leq \varepsilon$ yields

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) \approx \frac{1}{2\cosh(\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x}/2)}\varepsilon\|\boldsymbol{\theta}\|_*, \tag{3.12}$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$, which is defined as $\|\mathbf{z}\|_* \doteq \sup\{|\mathbf{z}^{\mathsf{T}}\mathbf{w}| : \|\mathbf{w}\| \leq 1\}$. Therefore, in this case, we obtain a weighted dual norm regularization on $\boldsymbol{\theta}$, with the weighting being large when $\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x}$ is close to zero (i.e., points with large uncertainty in the class assignment), and being small when $|\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x}|$ is large (i.e., points with low uncertainty in the class assignment). An even more direct connection between the FRD and the dual norm regularization on $\boldsymbol{\theta}$ can be obtained from Proposition 1, which leads to

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) \leq \frac{1}{2}|\boldsymbol{\delta}^{\mathsf{T}}\boldsymbol{\theta}| \leq \frac{1}{2}\varepsilon\|\boldsymbol{\theta}\|_*. \tag{3.13}$$

Equation 3.13 formalizes our intuitive idea that ℓ_p-regularized classifiers tend to be more robust. This is in agreement with other results, e.g., TORKAMANI and LOWD [2014].

3.3.4 FRD for the case of multiclass classification

Consider the general M-classification problem in which $\mathscr{Y} = \{1, \ldots, \mathrm{M}\}$, and let

$$q_{\boldsymbol{\theta}}(y|\mathbf{x}) = \frac{e^{h_y(\mathbf{x},\boldsymbol{\theta})}}{\sum_{y' \in \mathscr{Y}} e^{h_{y'}(\mathbf{x},\boldsymbol{\theta})}}, \tag{3.14}$$

be a standard softmax output, where $\mathbf{h} : \mathscr{X} \times \Theta \to \mathbb{R}^{\mathrm{M}}$ is a parametric representation function and z_y denotes the y-th component of the vector $\mathbf{z}$. The FRD for this model can also be obtained in closed-form as summarized below. The proof is given in Subsection 3.6.3.

Theorem 2 (FRD multiclass classifier)**.** *The* FRD *between soft-predictions* $q_{\boldsymbol{\theta}} \equiv q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ *and* $q'_{\boldsymbol{\theta}} \equiv q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')$*, according to Equation 3.14 and corresponding to inputs* $\mathbf{x}$ *and* $\mathbf{x}'$*, is*

given by

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) = 2\arccos\left(\sum_{y\in\mathscr{Y}}\sqrt{q_{\boldsymbol{\theta}}(y|\mathbf{x})\,q_{\boldsymbol{\theta}}(y|\mathbf{x}')}\right). \tag{3.15}$$

Remark. Although not obvious, the FRD for the multiclass case (Equation 3.15) is indeed consistent with the FRD for the binary case (Equation 3.9), i.e., they are equal for the case M = 2 (for further details the reader is referred to Subsection 3.6.3).

3.3.5 Comparison between FRD and KL divergence

The Fisher-Rao distance (Equation 3.15) has some interesting connections with other distances and information divergences. We are particularly interested in its relation with the KL divergence, which is the adversarial regularization mechanism used in the TRADES method [ZHANG and collab., 2019]. The next theorem summarizes the mathematical connection between these quantities. The proof of this theorem is relegated to Subsection A.3.2.

Theorem 3 (Relation between FRD and KL divergence)**.** *The* FRD *between soft-predictions* $q_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ *and* $q'_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')$*, given by Equation 3.15 is related to the KL divergence through the inequality:*

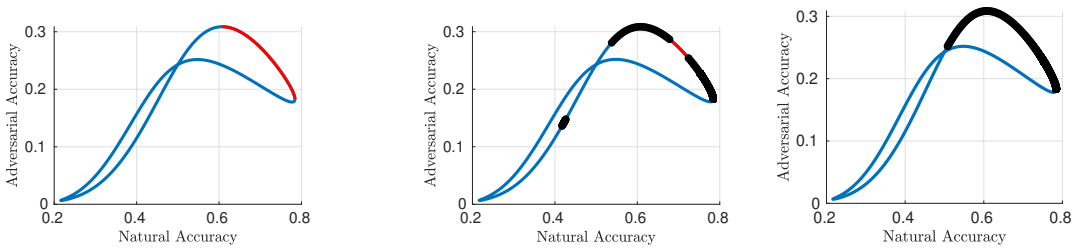
$$1 - \cos\left(\frac{d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}})}{2}\right) \leq \frac{1}{2}KL(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}), \tag{3.16}$$

which means that the KL divergence is a surrogate of the FRD*. In addition, it can also be shown that the KL divergence is a second-order approximation of the* FRD*, i.e.,*

$$KL(q_{\boldsymbol{\theta}} \| q'_{\boldsymbol{\theta}}) = \frac{1}{2}d^2_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) + \mathcal{O}(d^3_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}})), \tag{3.17}$$

where $\mathcal{O}(\cdot)$ *denotes big-O notation.*

The above result shows that the KL is a weak approximation of the FRD in the sense that it gives an upper bound and a second-order approximation of the geodesic distance. However, in general, we are interested in distances over arbitrarily distinct softmax distributions, so it is clear that the KL divergence and the FRD can behave very differently. In fact, only the latter measures the actual distance on the statistical manifold $\mathscr{C} \doteq \{q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) : \mathbf{x} \in \mathscr{X}\}$ (for further details, the reader is referred to Subsection 3.6.1). In the next section, we show that this has an important consequence on the set of solutions obtained by minimizing the respective empirical risks while varying λ.



(a) Pareto-optimal points (b) TRADES (c) FIRE

Figure 3.6: Plot of all the possible points $(1-\mathrm{P}_e(\boldsymbol{\theta}), 1-\mathrm{P}'_e(\boldsymbol{\theta}))$ for the Gaussian model with $\varepsilon = 0.1$, $\boldsymbol{\mu} = [-0.0218; 0.0425]$ and $\boldsymbol{\Sigma} = [0.0212, 0.0036; 0.0036, 0.0042]$ shown in blue. In red, we show the Pareto-optimal points (Figure 3.6a). In black, we show the solutions obtained by minimizing the risk $\mathrm{L}_{\mathrm{TRADES}}(\boldsymbol{\theta})$ in Equation 3.5 (Figure 3.6b), and the risk $\mathrm{L}_{\mathrm{FIRE}}(\boldsymbol{\theta})$ in Equation 3.7 (Figure 3.6c).©2022 IEEE.

3.4 Accuracy-Robustness Trade-offs and Learning in the Gaussian Model

To illustrate the FIRE loss and the role of the Fisher-Rao distance to encourage robustness, we study the natural-adversarial accuracy trade-off for a simple logistic regression and Gaussian model and compare the performance of the predictor trained on the FIRE loss with those of ACE and TRADES losses, in Equation 3.4 and Equation 3.5, respectively.

3.4.1 Accuracy-robustness trade-offs

Consider a binary example with a simplified logistic regression model. Therefore, in this section we assume that $\mathscr{Y} = \{-1, 1\}$ and the softmax probability

$$q_{\boldsymbol{\theta}}(y|\mathbf{x}) = \frac{1}{1+\exp(-y\,\boldsymbol{\theta}^{\mathsf{T}}\mathbf{x})}, \tag{3.18}$$

We choose the standard adversary obtained by maximizing the cross-entropy loss, i.e.,

$$\mathbf{x}'^{*} = \underset{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|\leq\varepsilon}{\operatorname{argmax}} \; -\log q_{\boldsymbol{\theta}}(y|\mathbf{x}'). \tag{3.19}$$

For simplicity[1], in this section we assume that the adversary uses the 2-norm (i.e, $\|\cdot\| = \|\cdot\|_2$). In such case, $\mathbf{x}'^{*}$ can be written as $\mathbf{x}'^{*} = \mathbf{x} - \varepsilon\, y\, \boldsymbol{\theta}/\|\boldsymbol{\theta}\|_2$.

We also assume that the classes are equally likely and that the conditional inputs given the class are Gaussian distributions with the particular form $\mathbf{x}|y \sim \mathcal{N}(y, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. In this case, we can write the natural and adversarial misclassification probabilities

[1] The analysis can be extended to an arbitrary norm but it is somewhat simplified for the 2-norm case.

as:

$$\mathrm{P}_e(\boldsymbol{\theta}) \doteq \mathbb{P}(f_{\boldsymbol{\theta}}(\mathbf{X}) \neq \mathrm{Y}) = \Phi\left(\frac{-\boldsymbol{\theta}^\intercal \boldsymbol{\mu}}{\sqrt{\boldsymbol{\theta}^\intercal \boldsymbol{\Sigma} \boldsymbol{\theta}}}\right), \tag{3.20}$$

$$\mathrm{P}'_e(\boldsymbol{\theta}) \doteq \mathbb{P}(f_{\boldsymbol{\theta}}(\mathbf{X}'^*) \neq \mathrm{Y}) = \Phi\left(\frac{\varepsilon\|\boldsymbol{\theta}\|_2 - \boldsymbol{\theta}^\intercal \boldsymbol{\mu}}{\sqrt{\boldsymbol{\theta}^\intercal \boldsymbol{\Sigma} \boldsymbol{\theta}}}\right), \tag{3.21}$$

where Φ denotes the cumulative distribution function of the standard normal random variable. The following result provides lower and upper bounds for $\mathrm{P}'_e(\boldsymbol{\theta})$ in terms of ε and the eigenvalues of $\boldsymbol{\Sigma}$. Notice that the bounds get sharper as ε or the spread of $\boldsymbol{\Sigma}$ decreases and are tight if $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$.

Proposition 2 (Accuracy-robustness trade-offs)**.** The adversarial misclassification probability $\mathrm{P}'_e(\boldsymbol{\theta})$ satisfies the inequalities:

$$\Phi\left(\frac{\varepsilon}{\lambda_{\max}^{1/2}(\boldsymbol{\Sigma})} + \Phi^{-1}(\mathrm{P}_e(\boldsymbol{\theta}))\right) \leq \mathrm{P}'_e(\boldsymbol{\theta}), \tag{3.22}$$

$$\Phi\left(\frac{\varepsilon}{\lambda_{\min}^{1/2}(\boldsymbol{\Sigma})} + \Phi^{-1}(\mathrm{P}_e(\boldsymbol{\theta}))\right) \geq \mathrm{P}'_e(\boldsymbol{\theta}). \tag{3.23}$$

The proof of this proposition is relegated to Subsection 3.6.4.

3.4.2 Learning

Let us consider the 2-dimensional case, i.e., $n = 2$. From Equation 3.20 and Equation 3.21, it can be noticed that both $\mathrm{P}_e(\boldsymbol{\theta})$ and $\mathrm{P}'_e(\boldsymbol{\theta})$ are independent of the 2-norm of $\boldsymbol{\theta}$. Therefore, we can parameterize $\boldsymbol{\theta}$ as $\boldsymbol{\theta} = [\cos(\alpha), \sin(\alpha)]^\intercal$ with $\alpha \in [0, 2\pi)$ without any loss of generality. Thus, $(1 - \mathrm{P}_e(\boldsymbol{\theta}), 1 - \mathrm{P}'_e(\boldsymbol{\theta}))$ for all values of α gives a curve that represents all the possible values of the natural and adversarial accuracies for this setting. Figure 3.6a shows this curve for a particular choice[2] of ε, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$. As can be observed, the solution which maximizes the natural accuracy gives poor adversarial accuracy and viceversa[3]. The set of Pareto-optimal points (i.e., the set of points for which there is no possible improvement in terms of both natural and adversarial accuracy or, equivalently, the set $\{\max_{\boldsymbol{\theta}} \beta(1 - \mathrm{P}_e(\boldsymbol{\theta})) + (1 - \beta)(1 - \mathrm{P}'_e(\boldsymbol{\theta})) : 0 \leq \beta \leq 1\}$) are also shown in Figure 3.6a. In particular, this set contains the Maximum Average Accuracy (MAA) given by

$$\mathrm{MAA} \doteq \max_{\boldsymbol{\theta} \in \Theta} \left(1 - \frac{\mathrm{P}_e(\boldsymbol{\theta}) + \mathrm{P}'_e(\boldsymbol{\theta})}{2}\right). \tag{3.24}$$

[2] In this experiment, we obtained the components in $\boldsymbol{\mu}$ by sampling $\mathcal{N}(0, 1/400)$. We also defined a matrix $\mathbf{A}$ with samples of the same distribution and constructed $\boldsymbol{\Sigma}$ as $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^\intercal$.

[3] The (normalized) value of $\boldsymbol{\theta}$ that maximizes $1 - \mathrm{P}_e(\boldsymbol{\theta})$ is $\theta^*_{\mathrm{nat}} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}/\|\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\|_2$ (see, for instance, Bishop [2006]) and corresponds to $\alpha^*_{\mathrm{nat}} \approx 1.814$, giving $1 - \mathrm{P}_e(\boldsymbol{\theta}) \approx 0.784$ and $1 - \mathrm{P}'_e(\boldsymbol{\theta}) \approx 0.183$. The value of θ that maximizes $1 - \mathrm{P}'_e(\boldsymbol{\theta})$ is $\alpha^*_{\mathrm{adv}} \approx 2.783$, giving $1 - \mathrm{P}_e(\boldsymbol{\theta}) \approx 0.608$ and $1 - \mathrm{P}'_e(\boldsymbol{\theta}) \approx 0.309$.

Table 3.1: Comparison between KL and Fisher-Rao based regularizer under white-box l_∞ threat model. Note that we do not use the same hyperparameters as presented in ZHANG and collab. [2019] for the TRADES method. ©2022 IEEE.

Defense	Dataset	ε	Structure	Natural	AutoAttack	Avg. Acc.	RunTime
TRADES	MNIST	0.3	CNN	**99.27 ± 0.03**	94.27 ± 0.18	96.77 ± 0.09	2h22
FIRE			CNN	99.22 ± 0.02	**94.44 ± 0.14**	**96.83 ± 0.10**	**2h06**
TRADES	CIFAR-10	8/255	WRN-34-10	85.84 ± 0.31	50.47 ± 0.36	68.15 ± 0.23	13h49
FIRE			WRN-34-10	**85.98 ± 0.09**	**51.45 ± 0.32**	**68.72 ± 0.22**	**11h00**
TRADES	CIFAR-100	8/255	WRN-34-10	59.62 ± 0.42	25.89 ± 0.26	42.76 ± 0.26	13h49
FIRE			WRN-34-10	**61.03 ± 0.21**	**26.42 ± 0.21**	**43.73 ± 0.12**	**11h10**

This is a metric of particular importance, which combines with equal weights both natural and adversarial accuracies.

In addition, we present the solution of the (local) Empirical Risk Minimization (ERM)[4] for the TRADES risk function as defined in Equation 3.5 for different values of λ in Figure 3.6b. As can be seen, the curve obtained in the $(1-\mathrm{P}_e(\boldsymbol{\theta}), 1-\mathrm{P}'_e(\boldsymbol{\theta}))$ space covers a large part of the Pareto-optimal points expect for a segment for which the solution does not behave well. Finally, in Figure 3.6c we present the result for the proposed FIRE risk function in Equation 3.7 for different values of λ. In this case, we observed that the curve of solutions in the $(1-\mathrm{P}_e(\boldsymbol{\theta}), 1-\mathrm{P}'_e(\boldsymbol{\theta}))$ space covers all the Pareto-optimal points. Moreover, we have observed that, for some λ, the $\boldsymbol{\theta}$ which minimizes the FIRE risk matches the $\boldsymbol{\theta}$ which achieves the MAA defined in Equation 3.24, while TRADES method fails in achieving this particularly relevant point. It should be added that none of the methods cover exactly the set of Pareto-optimal points, which is expected, since all loss functions can be considered surrogates for the quantity $\beta\mathrm{P}_e(\boldsymbol{\theta}) + (1-\beta)\mathrm{P}'_e(\boldsymbol{\theta})$, where $0 \leq \beta \leq 1$. We performed a similar comparison between FRD and KL using standard datasets. The results are reported in Appendix Section A.1

[4] For experiments, we used 10^4 samples for each different class and a BFGS Quasi-Newton method for optimization. The initial value of $\boldsymbol{\theta}$ is zero for both TRADES and FIRE. We do not report the result using ACE risk in Equation 3.4 because in this setting we would obtain the trivial solution $\boldsymbol{\theta} = \mathbf{0}$, which is a minimizer of $\mathrm{L}_{\mathrm{ACE}}(\boldsymbol{\theta})$.

Table 3.2: Test robustness on different datasets under white-box l_∞ attack. We ran all methods on 5 different tries and reported the mean and standard deviation. The codes for UAT and Atzmon et al. are not publicly available. Note that retraining the SOTA methods modifies slightly the experimental results. ©2022 IEEE.

Defense	Dataset	ε	Structure	Natural	AutoAttack	Avg. Acc.	Runtime
Without Additional Data							
Madry et al. [MADRY and collab., 2018]	MNIST	0.3	CNN	98.53 ± 0.06	88.62 ± 0.23	93.58 ± 0.14	**2h03**
Atzmon et al.[ATZMON and collab., 2019]			CNN	**99.35**	90.85	95.10	-
TRADES [ZHANG and collab., 2019]			CNN	99.27 ± 0.03	94.27 ± 0.18	96.77 ± 0.09	2h22
FIRE			CNN	99.22 ± 0.02	**94.44 ± 0.14**	**96.83 ± 0.10**	2h06
Madry et al. [MADRY and collab., 2018]	CIFAR-10	8/255	WRN-34-10	**87.56± 0.09**	44.07 ± 0.27	65.82 ± 0.15	**10h51**
Self Adaptive[HUANG and collab., 2020]			WRN-34-10	83.39 ± 0.19	53.11 ± 0.29	68.25 ± 0.14	13h57
TRADES [ZHANG and collab., 2019]			WRN-34-10	84.79 ± 0.24	52.12± 0.28	68.45 ± 0.12	17h49
FIRE + Self Adaptive			WRN-34-10	83.70 ± 0.36	**53.26 ± 0.19**	68.48 ± 0.13	11h12
Overfitting [RICE and collab., 2020]			WRN-34-10	85.64 ± 0.55	51.72 ± 0.56	68.68 ± 0.44	42h01
FIRE			WRN-34-10	85.98 ± 0.09	51.45 ± 0.32	**68.72 ± 0.22**	11h00
Overfitting [RICE and collab., 2020]	CIFAR-100	8/255	RN-18	53.83	18.95	36.39	-
Overfitting [RICE and collab., 2020]			WRN-34-10	59.22 ± 0.61	25.99 ± 0.51	42.61 ± 0.28	42h08
FIRE			WRN-34-10	**61.03 ± 0.21**	**26.42 ± 0.21**	**43.73 ± 0.12**	**11h10**
With Additional Data Using 80M-TI							
Pre-training[HENDRYCKS and collab., 2019]	CIFAR-10	8/255	WRN-28-10	86.93 ± 0.79	53.35 ± 0.81	70.14 ± 0.54	40h00 + 0h20
UAT [ALAYRAC and collab., 2019]			WRN-106-8	86.46	56.03	71.24	-
MART [WANG and collab., 2019]			WRN-28-10	87.39 ± 0.12	56.69 ± 0.28	72.04 ± 0.15	**13h53**
RST-adv [CARMON and collab., 2019]			WRN-28-10	89.49 ± 0.41	59.69 ± 0.26	74.59 ± 0.27	22h12
FIRE			WRN-28-10	**89.73 ± 0.04**	**59.97 ± 0.11**	**74.86 ± 0.05**	18h30

3.5 Experimental Results

In this section, we assess our proposed FIRE loss' effectiveness to improve neural networks' robustness.[5]

3.5.1 Setup

Datasets: We resort to standard benchmarks. First, we use MNIST [LECUN and collab., 2010], composed of 60,000 black and white images of size 28×28 - 50,000 for training, and 10,000 for testing - divided into 10 different classes. Then, we test on CIFAR-10, and CIFAR-100 [KRIZHEVSKY, 2009], composed of 60,000 color images of size 32×32×3 - 50,000 for training and 10,000 for testing - divided into 10 and 100 classes, respectively. Finally, we also test on CIFAR-10 with additional data thanks to 80 Million Tiny Images [TORRALBA and collab., 2008], an experiment that will be detailed later.

Architectures: In order to provide fair comparisons, we use standard model architectures. For the MNIST simulations, we use the 7-layer CNN as in ZHANG and collab. [2019]. For CIFAR-10 and CIFAR-100, we use a WideResNet (WRN) with 34 layers, and a widen factor of 10 (shortened as WRN-34-10) as in MADRY and collab. [2018]; WU and collab. [2020]; ZHANG and collab. [2019]. For the simulations with additional data, we use a WRN-28-10 as in CARMON and collab. [2019].

Training procedure: For all standard experiments (without additional data), we use a Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9, a weight decay of $5 \cdot 10^{-4}$ and Nesterov momentum. We train our models on 100 epochs with a batch size equal to 256. The initial learning rate is set to 0.01 for MNIST and 0.1

[5] Codes are available on GitHub at https://github.com/MarinePICOT/Adversarial-Robustness-via-Fisher-Rao-Regularization

for CIFAR-10 and CIFAR-100. Following ZHANG and collab. [2019], the learning rate is divided by 10 at epochs 75 and 90. For our experiments with additional data, we follow the protocol introduced by CARMON and collab. [2019], using a cosine decay [LOSHCHILOV and HUTTER, 2017], and training on 200 epochs.

Generation of adversarial samples: For all experiments, we use PGD [MADRY and collab., 2018] to generate the adversarial examples during training. The loss which is maximized during the PGD algorithm is the Fisher-Rao distance (FRD) for our experiments, and the Kullback-Leibler divergence for the TRADES method. For MNIST, we use 40 steps and 10 steps for the rest. The step size is set to 0.01 for MNIST and 0.007 for the others. The maximal distortion in l_∞-norm ϵ allowed is 0.031 for CIFAR-10 and CIFAR-100, and 0.3 for MNIST. This setting is used in most methods, such as CARMON and collab. [2019]; GOODFELLOW and collab. [2015].

Additional data: To improve performance, CARMON and collab. [2019] propose the use of additional data when training on CIFAR-10. Specifically, CARMON and collab. [2019] use 500k additional images from 80M-TI [6]. Those images have been selected such that their l_2-distance to images from CIFAR-10 are below a threshold.

Hyperparameters: The Rao regularizer used to improve the robustness of neural networks introduces a hyperparameter λ to balance natural accuracy and adversarial robustness. We select $\lambda = 12$ from the CIFAR-10 simulations and use this value for all the other datasets. Further study on the effect of λ is provided in Section 3.5.2.

Test metrics: First, we provide the accuracy of the model on clean samples after adversarial training (Natural). Second, we test our models using the recently introduced AutoAttack [CROCE and HEIN, 2020], keeping the same hyperparameters as the ones used in the original paper and code[7]. AutoAttack tests the model under a comprehensive series of attacks and provides a more reliable assessment of robustness than the traditionally used PGD-based evaluation. Given that we care equally about natural and adversarial accuracies, we also compute the average sum of the two, i.e., the Average Accuracy (Avg. Acc.). This is an empirical version of the Maximum Average Accuracy (MAA) defined in Equation 3.24. Finally, we report the runtime of each method as the time required to complete the adversarial training. To provide fair comparisons between runtimes, we run the official code of each method on the same 4 NvidiaV100 GPUs (for further details, see Section 3.5.2).

3.5.2 Experimental results

Kullback-Leibler versus Fisher-Rao regularizer:

To disentangle the influence of different regularizers, we compare the Fisher-Rao-based regularizer to the Kullback-Leibler-based regularizer used in TRADES with

[6] Images available at https://github.com/yaircarmon/semisup-adv

[7] The AutoAttack code is available on https://github.com/fra31/auto-attack

the exact same model and hyperparameters (as detailed previously) on CIFAR-10, CIFAR-100, and MNIST datasets. We used $\lambda = 6$ to train the TRADES method, since it is the value presented in the original paper [ZHANG and collab., 2019]. The results are averaged over 5 tries and summarized in Table 3.1. Those results confirm the superiority of the proposed regularizer. Specifically, the natural and adversarial accuracies increase up to 1% each under AutoAttack, improving the trade-off up to 1%.

Comparison with state-of-the-art

We compare FIRE with the state-of-the-art methods using adversarial training under l_∞-norm attacks. Due to its effectiveness on adversarial performances, we use the self-adaptive scheme from HUANG and collab. [2020] along with the FIRE loss to smooth the one-hot labels on CIFAR-10, and report the results under FIRE + Self Adaptive. We do not include the Adversarial Weight Perturbation (AWP) [WU and collab., 2020] method since it leverages two networks to increase the robustness of the model. We trained all methods from the state-of-the art on 5 tries and report the mean the standard variance of the results in Table 3.2. Note that the codes for UAT and Atzmon et al. are not publicly available, therefore we reported the results available on RobustBench [CROCE and collab., 2020].

Average Accuracy (Avg. Acc.): Overall, FIRE exhibits the best Avg. Acc. among compared methods in all settings. On CIFAR-10, at equivalent method, our FIRE method outperforms HUANG and collab. [2020]. The gain on natural examples is close to 0.3% and 0.15% in adversarial performances, giving an improvemnt of 0.23% of Avg. Acc. Moreover, our FIRE method performs slightly better (0.04%) than RICE and collab. [2020] but, given that their method requires four times FIRE's runtime to complete its training, the gain of our method appears to be more significant. Besides, FIRE outperforms RICE and collab. [2020] on the more challenging CIFAR-100 in both Avg. Acc., with more significant gain (1.12%) and runtime. Taking all metrics into account, FIRE appears to be the best overall method.

Runtimes: Interestingly, our method exhibits a significant advantage over previous state-of-the-art methods using similar backbones. Our method outruns methods with similar performances by 20% on average. We presume that the difference between the different runtimes comes from the fact that our proposed FIRE loss comprises 5 different operations. In contrast, the KL divergence, proposed in ZHANG and collab. [2019], is composed of 6 operations.

Ablation studies

Influence of λ: We study the influence of the hyperparameter λ on the performances of the FIRE method. Figure 3.7 clearly shows the trade-off between natural and ad-

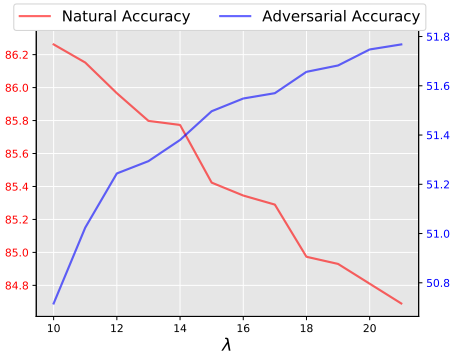


Figure 3.7: Influence of the hyperparameter λ on the natural and adversarial accuracies for FIRE regularizer on CIFAR-10. ©2022 IEEE.

versarial accuracy under AutoAttack [CROCE and HEIN, 2020]. When increasing λ, we emphasize our robust regularizer, consequently decreasing the performance on clean samples. Such phenomenon properly aligns with intuition and is also observed in ZHANG and collab. [2019]. Even though several values of λ lead to reasonable performances, we chose λ = 12 to have a natural accuracy close to the natural accuracy under the TRADES method en CIFAR-10. This ensures a fair comparison.

3.6 Proofs of Theorems and Propositions

3.6.1 Review of Fisher-Rao Distance (FRD)

Consider the family of probability distributions over the classes[8] $\mathscr{C} \doteq \{q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) : \mathbf{x} \in \mathscr{X}\}$. Assume that the following regularity conditions hold [ATKINSON and MITCHELL, 1981]:

(i) $\nabla_{\mathbf{x}} q_{\boldsymbol{\theta}}(y|\mathbf{x})$ exists for all $\mathbf{x}$, y and $\boldsymbol{\theta} \in \Theta$;

(ii) $\sum_{y \in \mathscr{Y}} \nabla_{\mathbf{x}} q_{\boldsymbol{\theta}}(y|\mathbf{x}) = 0$ for all $\mathbf{x}$ and $\boldsymbol{\theta} \in \Theta$;

(iii) $\mathbf{G}(\mathbf{x}) = \mathbb{E}_{\mathrm{Y} \sim q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})} \left[\nabla_{\mathbf{x}} \log q_{\boldsymbol{\theta}}(\mathrm{Y}|\mathbf{x}) \nabla_{\mathbf{x}}^{\mathsf{T}} \log q_{\boldsymbol{\theta}}(\mathrm{Y}|\mathbf{x}) \right]$ is positive definite for any $\mathbf{x}$ and $\boldsymbol{\theta} \in \Theta$.

Notice that if (i) holds, (ii) also holds immediately for discrete distributions over finite spaces (assuming that $\sum_{y \in \mathscr{Y}}$ and $\nabla_{\mathbf{x}}$ are interchangeable operations) as in our case. When (i)-(iii) are met, the variance of the differential form $\nabla_{\mathbf{x}}^{\mathsf{T}} \log q_{\boldsymbol{\theta}}(\mathrm{Y}|\mathbf{x}) d\mathbf{x}$ can be interpreted as the square of a differential arc length ds^2 in the space $\mathscr{C}$, which yields

$$ds^2 = \langle d\mathbf{x}, d\mathbf{x} \rangle_{\mathbf{G}(\mathbf{x})} = d\mathbf{x}^{\mathsf{T}} \mathbf{G}(\mathbf{x}) d\mathbf{x}. \tag{3.25}$$

[8] Since we are interested in the dependence of $q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ with changes in input $\mathbf{x}$ and, particularly, its robustness to adversarial perturbations, we consider $\mathbf{x}$ as the "parameters" of the model over which the regularity conditions must be imposed.

Thus, $\mathbf{G}$, which is the Fisher Information Matrix (FIM), can be adopted as a metric tensor. We now consider a curve $\boldsymbol{\gamma} : [0,1] \to \mathscr{X}$ in the input space connecting two arbitrary points $\mathbf{x}$ and $\mathbf{x}'$, i.e., such that $\boldsymbol{\gamma}(0) = \mathbf{x}$ and $\boldsymbol{\gamma}(1) = \mathbf{x}'$. Notice that this curve induces the following curve in the space $\mathscr{C}$: $q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{\gamma}(t))$ for $t \in [0,1]$. The Fisher-Rao distance between the distributions $q_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ and $q'_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')$ will be denoted as $d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}})$ and is formally defined by

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) \doteq \inf_{\boldsymbol{\gamma}} \int_0^1 \sqrt{\frac{d\boldsymbol{\gamma}^\intercal(t)}{dt} \mathbf{G}(\boldsymbol{\gamma}(t)) \frac{d\boldsymbol{\gamma}(t)}{dt}}, \tag{3.26}$$

where the infimum is taken over all piecewise smooth curves. This means that the FRD is the length of the *geodesic* between points $\mathbf{x}$ and $\mathbf{x}'$ using the FIM as the metric tensor. In general, the minimization of the functional in Equation 3.26 is a problem that can be solved using the well-known Euler-Lagrange differential equations. Several examples for simple families of distributions can be found in ATKINSON and MITCHELL [1981].

3.6.2 Proof of Theorem 1

To compute the FRD, we first need to compute the FIM of the family $\mathscr{C}$ with $q_{\boldsymbol{\theta}}(y|\mathbf{x})$ given in Equation 3.8. A direct calculation gives:

$$\mathbf{G}(\mathbf{x}) = \frac{e^{h_{\boldsymbol{\theta}}(\mathbf{x})}}{\left(1 + e^{h_{\boldsymbol{\theta}}(\mathbf{x})}\right)^2} \nabla_{\mathbf{x}} h_{\boldsymbol{\theta}}(\mathbf{x}) \nabla_{\mathbf{x}}^\intercal h_{\boldsymbol{\theta}}(\mathbf{x}). \tag{3.27}$$

It is clear from this expression that the FIM of this model is of rank one and therefore singular. This matches the fact that $q_{\boldsymbol{\theta}}(y|\mathbf{x})$ has a single degree of freedom given by $h_{\boldsymbol{\theta}}(\mathbf{x})$. Therefore, the statistical manifold $\mathscr{C}$ has dimension 1.

To proceed, we consider the following model:

$$q_{\boldsymbol{\theta}}(y|u) = \frac{1}{1 + e^{-uy}}, \tag{3.28}$$

where we have defined $u = h_{\boldsymbol{\theta}}(\mathbf{x})$. This effectively removes the model ambiguities because $q(y|u) \neq q(y|u')$ if and only if $u \neq u'$. Note that $q_{\boldsymbol{\theta}}(y|u)$, for fixed $\boldsymbol{\theta}$, can be interpreted as a one-dimensional parametric model with parameter u. Its FIM is a scalar that can be readily obtained, yielding:

$$\mathrm{G}(u) = \frac{e^u}{(1 + e^u)^2}. \tag{3.29}$$

Clearly, the FIM $\mathrm{G}(u)$ is non-singular for any u (i.e., $\mathrm{G}(u) > 0$ for any u) as required. Let $\mathscr{D} = \{q_{\boldsymbol{\theta}}(\cdot|u) : u \in \mathscr{U}\}$, where $\mathscr{U} = h_{\boldsymbol{\theta}}(\mathscr{X})$ and consider two distributions in $\mathscr{D}$: $q_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|u)$ and $q'_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|u')$. Then, the FRD can be evaluated directly as follows

[ATKINSON and MITCHELL, 1981][Eq. (3.13)]:

$$d_{\mathrm{R},\mathscr{D}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) = \left| \int_u^{u'} \mathrm{G}^{1/2}(v)\, dv \right| = 2 \left| \arctan\left(e^{u'/2}\right) - \arctan\left(e^{u/2}\right) \right|. \tag{3.30}$$

Therefore, the FRD between the distributions $q_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ and $q'_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')$ can be directly obtained by substituting $u = h_{\boldsymbol{\theta}}(\mathbf{x})$ and $u' = h_{\boldsymbol{\theta}}(\mathbf{x}')$ in Equation 3.30, yielding the final result in Equation 3.9.

3.6.3 Proof of Theorem 2

As in the binary case developed in Subsection 3.3.3, the FIM of the family $\mathscr{C}$ with $q_{\boldsymbol{\theta}}(y|\mathbf{x})$ given in Equation 3.14 is singular. To shows this, we first notice that $q_{\boldsymbol{\theta}}(y|\mathbf{x})$ can be written as

$$q_{\boldsymbol{\theta}}(y|\mathbf{x}) = \frac{e^{g_y(\mathbf{x},\boldsymbol{\theta})}}{\sum_{y' \in \mathscr{Y}} e^{g_{y'}(\mathbf{x},\boldsymbol{\theta})}}, \tag{3.31}$$

where $g_y(\mathbf{x},\boldsymbol{\theta}) = h_y(\mathbf{x},\boldsymbol{\theta}) - h_1(\mathbf{x},\boldsymbol{\theta})$. Since $g_1(\mathbf{x},\boldsymbol{\theta}) = 0$, this shows that $\mathscr{C}$ has $\mathrm{M}-1$ degrees of freedom: $g_2(\mathbf{x},\boldsymbol{\theta}), \ldots, g_{\mathrm{M}}(\mathbf{x},\boldsymbol{\theta})$. A direct calculation of the FIM gives

$$\begin{aligned} \mathbf{G}(\mathbf{x}) = &\sum_{y=2}^{\mathrm{M}} q_{\boldsymbol{\theta}}(y|\mathbf{x}) \nabla_{\mathbf{x}} g_y(\mathbf{x},\boldsymbol{\theta}) \nabla_{\mathbf{x}}^{\mathsf{T}} g_y(\mathbf{x},\boldsymbol{\theta}) \\ &- \sum_{y,y'=2}^{\mathrm{M}} q_{\boldsymbol{\theta}}(y|\mathbf{x}) q_{\boldsymbol{\theta}}(y'|\mathbf{x}) \nabla_{\mathbf{x}} g_y(\mathbf{x},\boldsymbol{\theta}) \nabla_{\mathbf{x}} g_{y'}^{\mathsf{T}}(\mathbf{x},\boldsymbol{\theta}). \end{aligned} \tag{3.32}$$

Let $\mathbf{v} \in \mathscr{X}$ be an arbitrary vector and define $\beta_y \doteq \nabla_{\mathbf{x}}^{\mathsf{T}} g_y(\mathbf{x},\boldsymbol{\theta})\, \mathbf{v}$ for $y = [2:\mathrm{M}]$. Notice that

$$\begin{aligned} \mathbf{G}(\mathbf{x})\mathbf{v} = &\sum_{y=2}^{\mathrm{M}} q_{\boldsymbol{\theta}}(y|\mathbf{x}) \beta_y \nabla_{\mathbf{x}} g_y(\mathbf{x},\boldsymbol{\theta}) \\ &- \sum_{y=2}^{\mathrm{M}} \left(\sum_{y'=2}^{\mathrm{M}} q_{\boldsymbol{\theta}}(y'|\mathbf{x}) \beta_{y'} \right) q_{\boldsymbol{\theta}}(y|\mathbf{x}) \nabla_{\mathbf{x}} g_y(\mathbf{x},\boldsymbol{\theta}). \end{aligned} \tag{3.33}$$

Therefore, the range of $\mathbf{G}(\mathbf{x})$ is a subset of the span of the set $\{\nabla_{\mathbf{x}} g_2(\mathbf{x},\boldsymbol{\theta}), \ldots, \nabla_{\mathbf{x}} g_{\mathrm{M}}(\mathbf{x},\boldsymbol{\theta})\}$. Thus, it follows that $\mathrm{rank}(\mathbf{G}(\mathbf{x})) \leq \mathrm{M}-1$, which implies that it is singular.

The singularity issue can be overcome by embedding $\mathscr{C}$ into the probability simplex $\mathscr{P}$ defined as follows:

$$\mathscr{P} = \left\{ q : \mathscr{Y} \to [0,1]^{\mathrm{M}} : \sum_{y \in \mathscr{Y}} q(y) = 1 \right\}. \tag{3.34}$$

To proceed, we follow CALIN and UDRIŞTE [2014][Section 2.8] and consider the

following parameterization for any distribution $q \in \mathscr{P}$:

$$q(y|\mathbf{z}) = \frac{z_y^2}{4}, \quad y \in \{1, \dots, \mathrm{M}\}. \tag{3.35}$$

We then consider the following statistical manifold:

$$\mathscr{D} = \left\{ q(\cdot|\mathbf{z}) : \|\mathbf{z}\|^2 = 4, z_y \geq 0, \ \forall\, y \in \mathscr{Y} \right\}. \tag{3.36}$$

Notice that the parameter vector $\mathbf{z}$ belongs to the positive portion of a sphere of radius 2 and centered at the origin in $\mathbb{R}^{\mathrm{M}}$. As a consequence, the FIM follows by

$$\mathbf{G}(\mathbf{z}) = \sum_{y \in \mathscr{Y}} \frac{z_y^2}{4} \left(\frac{2}{z_y} \mathbf{e}_y \right) \left(\frac{2}{z_y} \mathbf{e}_y^{\mathsf{T}} \right) = \mathbf{I}, \tag{3.37}$$

where $\{\mathbf{e}_y\}$ are the canonical basis vectors in $\mathbb{R}^{\mathrm{M}}$ and $\mathbf{I}$ is the identity matrix. From Equation 3.37 we can conclude that the Fisher metric is equal to the Euclidean metric. Since the parameter vector lies on a sphere, the FRD between the distributions $q = q(\cdot|\mathbf{z})$ and $q' = q(\cdot|\mathbf{z}')$ can be written as the radius of the sphere times the angle between the vectors $\mathbf{z}$ and $\mathbf{z}'$. This leads to

$$\begin{aligned} d_{\mathrm{R},\mathscr{D}}(q, q') &= 2 \arccos\left(\frac{\mathbf{z}^{\mathsf{T}} \mathbf{z}'}{4} \right) \\ &= 2 \arccos\left(\sum_{y \in \mathscr{Y}} \sqrt{q(y|\mathbf{z}) q(y|\mathbf{z}')} \right). \end{aligned} \tag{3.38}$$

Finally, we can compute the FRD for distributions in $\mathscr{C}$ using:

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) = 2 \arccos\left(\sum_{y \in \mathscr{Y}} \sqrt{q_{\boldsymbol{\theta}}(y|\mathbf{x}) q_{\boldsymbol{\theta}}(y|\mathbf{x}')} \right). \tag{3.39}$$

Notice that $0 \leq d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) \leq \pi$, $\forall\, \mathbf{x}, \mathbf{x}' \in \mathscr{X}$, being zero if $q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')$ and π when $[q_{\boldsymbol{\theta}}(1|\mathbf{x}), \dots, q_{\boldsymbol{\theta}}(\mathrm{M}|\mathbf{x})]$ and $[q_{\boldsymbol{\theta}}(1|\mathbf{x}'), \dots, q_{\boldsymbol{\theta}}(\mathrm{M}|\mathbf{x}')]$ are orthogonal vectors.

Proof the consistency between Theorem 1 and Theorem 2. This consists in showing the equivalence between the FRD for the binary case, given by Equation 3.9, and the multiclass case, given by Equation 3.15, for $\mathrm{M} = 2$. First, notice that the models Equation 3.8 and Equation 3.14 coincide if we consider the following correspondence: $h_{\boldsymbol{\theta}}(\mathbf{x}) = h_2(\mathbf{x}, \boldsymbol{\theta}) - h_1(\mathbf{x}, \boldsymbol{\theta})$, $y = 1 \leftrightarrow y = -1$ and $y = 2 \leftrightarrow y = 1$. Then, using

standard trigonometric identities, we rewrite the FRD for the multiclass case:

$$
\begin{aligned}
d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) &= 2\arccos\left(\sqrt{\frac{1}{1+e^{h_{\boldsymbol{\theta}}(\mathbf{x})}}}\sqrt{\frac{1}{1+e^{h_{\boldsymbol{\theta}}(\mathbf{x}')}}}\right. \\
&\left.+\sqrt{\frac{1}{1+e^{-h_{\boldsymbol{\theta}}(\mathbf{x})}}}\sqrt{\frac{1}{1+e^{-h_{\boldsymbol{\theta}}(\mathbf{x}')}}}\right) \\
&= 2\arccos\Big[\cos(\arctan(e^{h_{\boldsymbol{\theta}}(\mathbf{x})/2}))\cos(\arctan(e^{h_{\boldsymbol{\theta}}(\mathbf{x}')/2})) \\
&+\sin(\arctan(e^{h_{\boldsymbol{\theta}}(\mathbf{x})/2}))\sin(\arctan(e^{h_{\boldsymbol{\theta}}(\mathbf{x}')/2}))\Big] \\
&= 2\arccos\Big[\cos\Big(\arctan(e^{h_{\boldsymbol{\theta}}(\mathbf{x})/2}) - \arctan(e^{h_{\boldsymbol{\theta}}(\mathbf{x}')/2})\Big)\Big] \\
&= 2\Big|\arctan(e^{h_{\boldsymbol{\theta}}(\mathbf{x}')/2}) - \arctan(e^{h_{\boldsymbol{\theta}}(\mathbf{x})/2})\Big|,
\end{aligned}
\tag{3.40}
$$

where the last step follows by $|\arctan(\alpha)| \leq \pi/2$, $\forall \alpha \in \mathbb{R}$, so the argument of the cosine function belongs to $[-\pi, \pi]$ and $\arccos(\cos(\alpha)) = |\alpha|$ for $|\alpha| \leq \pi$, which completes the proof.

3.6.4 Proof of Proposition 2

For completeness, we first present the derivation of the misclassification probabilities Equation 3.20 and Equation 3.21:

$$
\mathrm{P}_e(\boldsymbol{\theta}) = \mathbb{P}(f_{\boldsymbol{\theta}}(\mathbf{X}) \neq \mathrm{Y}) = \Phi\left(-\frac{\theta^\intercal \boldsymbol{\mu}}{\sqrt{\theta^\intercal \boldsymbol{\Sigma} \theta}}\right), \tag{3.41}
$$

$$
\mathrm{P}'_e(\boldsymbol{\theta}) = \mathbb{P}(f_{\boldsymbol{\theta}}(\mathbf{X}'^*) \neq \mathrm{Y}) = \Phi\left(\frac{\varepsilon\|\boldsymbol{\theta}\|_2 - \theta^\intercal \boldsymbol{\mu}}{\sqrt{\theta^\intercal \boldsymbol{\Sigma} \theta}}\right). \tag{3.42}
$$

Notice that Φ, i.e., the cumulative distribution function of the standard normal random variable, is a monotonic increasing function and $\varepsilon\|\boldsymbol{\theta}\|_2/\sqrt{\theta^\intercal \boldsymbol{\Sigma} \theta} \geq 0$, so we have $\mathrm{P}'_e(\boldsymbol{\theta}) \geq \mathrm{P}_e(\boldsymbol{\theta})$, as expected. Furthermore, observe also that Φ is invertible so we can write $\mathrm{P}'_e(\boldsymbol{\theta})$ explicitly as a function of $\mathrm{P}_e(\boldsymbol{\theta})$:

$$
\mathrm{P}'_e(\boldsymbol{\theta}) = \Phi\left(\frac{\varepsilon\|\boldsymbol{\theta}\|_2}{\sqrt{\theta^\intercal \boldsymbol{\Sigma} \theta}} + \Phi^{-1}(\mathrm{P}_e)\right). \tag{3.43}
$$

We now proceed to the proof of the proposition. Notice that by the Rayleigh theorem [Horn and Johnson, 2013][Theorem 4.2.2], we have that

$$
\lambda_{\min}(\boldsymbol{\Sigma})\|\boldsymbol{\theta}\|_2^2 \leq \theta^\intercal \boldsymbol{\Sigma} \theta \leq \lambda_{\max}(\boldsymbol{\Sigma})\|\boldsymbol{\theta}\|_2^2, \tag{3.44}
$$

where $\lambda_{\min}(\boldsymbol{\Sigma})$ and $\lambda_{\max}(\boldsymbol{\Sigma})$ are the minimum and maximum eigenvalues of $\boldsymbol{\Sigma}$, respectively. Therefore, we can bound $\varepsilon\|\boldsymbol{\theta}\|_2/\sqrt{\theta^\intercal \boldsymbol{\Sigma} \theta}$ as follows:

$$
\frac{\varepsilon}{\lambda_{\max}^{1/2}(\boldsymbol{\Sigma})} \leq \frac{\varepsilon\|\boldsymbol{\theta}\|_2}{\sqrt{\theta^\intercal \boldsymbol{\Sigma} \theta}} \leq \frac{\varepsilon}{\lambda_{\min}^{1/2}(\boldsymbol{\Sigma})}. \tag{3.45}
$$

Using this fact together with the monotonicity of Φ in Equation 3.43, we obtain the desired result.

3.7 Summary and Concluding Remarks

We introduced FIRE, a new robustness regularizer-based method on the geodesic distance of softmax probabilities using concepts from information geometry. The main innovation is to employ Fisher-Rao Distance (FRD) to encourage invariant softmax probabilities for both natural and adversarial examples while maintaining high performances on natural samples. Our empirical results showed that FIRE consistently enhances the robustness of neural networks using various architectures, settings, and datasets. Compared to the state-of-the-art methods for adversarial defenses, FIRE increases the Average Accuracy (Avg. Acc.). Besides, it succeeds in doing so with a 20% reduction in terms of the training time.

Interestingly, FRD has rich connections with Hellinger distance, the Kullback-Leibler divergence, and even other standard regularization terms. Moreover, as illustrated via our simple logistic regression and Gaussian model, the optimization based on FIRE is well-behaved and gives all the desired Pareto-optimal points in the natural-adversarial region. This observation contrasts with the results of other state-of-the-art adversarial learning approaches. Further theoretical explanation of this change in behaviour is worth exploring in a future work.

Chapter 3 Conclusion

In this chapter, we addressed the problem of leveraging the knowledge about the output space to increase an image classifier's robustness. We introduced a new robustness regularizer-based method on the Fisher-Rao distance. The Fisher-Rao Distance (FRD) is the geodesic distance of softmax probabilities using concepts from information-geometry that will encourage invariant softmax probabilities for both natural and adversarial examples while maintaining high performances on natural samples. Our empirical results showed that using the Fisher-Rao distance consistently enhances the robustness of neural networks using various architectures, settings, and datasets.
Our framework is powerful and generic enough to be adapted to various applications. In what follows, we rely on the same idea to increase the safety of the cyber-components of Smart Grid systems. We focused on the protection of the state estimator, a component necessary to monitor the grid in real-time that is highly sensitive to attacks. In Chapter 4, we restrict ourselves to the linearized case, while in Chapter 5, we extend it in the non-linear setting.

3.8 References

ALAYRAC, J.-B., J. UESATO, P.-S. HUANG, A. FAWZI, R. STANFORTH and P. KOHLI. 2019, «Are labels required for improving adversarial robustness?», in *Advances in Neural Information Processing Systems*, p. 12 214–12 223. 93, 105

ALZANTOT, M., Y. SHARMA, A. ELGOHARY, B.-J. HO, M. SRIVASTAVA and K.-W. CHANG. «Generating natural language adversarial examples», in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 91

ATKINSON, C. and A. F. S. MITCHELL. 1981, «Rao's distance measure», *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, vol. 43, nº 3, p. 345–365, ISSN 0581572X. 108, 109, 110

ATZMON, M., N. HAIM, L. YARIV, O. ISRAELOV, H. MARON and Y. LIPMAN. 2019, «Controlling neural level sets», in *Advances in Neural Information Processing Systems*, p. 2034–2043. 93, 105

BISHOP, C. M. 2006, *Pattern Recognition and Machine Learning*, Springer. 103

CALIN, O. and C. UDRIŞTE. 2014, *Geometric modeling in probability and statistics*, Springer. 110

CARLINI, N. and D. WAGNER. 2017, «Adversarial examples are not easily detected: Bypassing ten detection methods», in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, ACM, New York, NY, USA, ISBN 978-1-4503-5202-4, p. 3–14, doi: 10.1145/3128572.3140444. 91

CARMON, Y., A. RAGHUNATHAN, L. SCHMIDT, J. C. DUCHI and P. S. LIANG. 2019, «Unlabeled data improves adversarial robustness», in *Advances in Neural Information Processing Systems*, p. 11 192–11 203. 93, 105, 106

CISSE, M., P. BOJANOWSKI, E. GRAVE, Y. DAUPHIN and N. USUNIER. 2017, «Parseval networks: Improving robustness to adversarial examples», in *Proceedings of*

the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 70, édité par D. Precup and Y. W. Teh, PMLR, p. 854–863. 96

COHEN, J., E. ROSENFELD and Z. KOLTER. 2019, «Certified adversarial robustness via randomized smoothing», in *International Conference on Machine Learning*, PMLR, p. 1310–1320. 92

CROCE, F., M. ANDRIUSHCHENKO and M. HEIN. 2019, «Provable robustness of relu networks via maximization of linear regions», in *the 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, p. 2057–2066. 92

CROCE, F., M. ANDRIUSHCHENKO, V. SEHWAG, E. DEBENEDETTI, N. FLAMMARION, M. CHIANG, P. MITTAL and M. HEIN. 2020, «Robustbench: a standardized adversarial robustness benchmark», *arXiv preprint arXiv:2010.09670*. 107

CROCE, F. and M. HEIN. 2020, «Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks», in *International Conference on Machine Learning*, PMLR, p. 2206–2216. 91, 106, 108

FEINMAN, R., R. R. CURTIN, S. SHINTRE and A. B. GARDNER. 2017, «Detecting adversarial samples from artifacts», *CoRR*, vol. abs/1703.00410. 91

GILMER, J., L. METZ, F. FAGHRI, S. S. SCHOENHOLZ, M. RAGHU, M. WATTENBERG and I. J. GOODFELLOW. 2018, «Adversarial spheres», . 91

GOODFELLOW, I. J., J. SHLENS and C. SZEGEDY. 2015, «Explaining and harnessing adversarial examples», *International Conference on Learning Representations*. 91, 92, 93, 106

GOWAL, S., K. D. DVIJOTHAM, R. STANFORTH, R. BUNEL, C. QIN, J. UESATO, R. ARANDJELOVIC, T. MANN and P. KOHLI. 2019, «Scalable verified training for provably robust image classification», in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, p. 4842–4851. 92

GROSSE, K., P. MANOHARAN, N. PAPERNOT, M. BACKES and P. D. MCDANIEL. 2017, «On the (statistical) detection of adversarial examples», *CoRR*, vol. abs/1702.06280. 91

HENDRYCKS, D., K. LEE and M. MAZEIKA. 2019, «Using pre-training can improve model robustness and uncertainty», in *International Conference on Machine Learning*, PMLR, p. 2712–2721. 93, 105

HINTON, G., O. VINYALS and J. DEAN. 2015, «Distilling the knowledge in a neural network», in *NIPS Deep Learning and Representation Learning Workshop*. 92

HORN, R. and C. JOHNSON. 2013, *Matrix Analysis*, Matrix Analysis, Cambridge University Press, ISBN 9780521839402. 112

HUANG, L., C. ZHANG and H. ZHANG. 2020, «Self-adaptive training: beyond empirical risk minimization», *Advances in Neural Information Processing Systems*, vol. 33. 93, 105, 107

ILYAS, A., S. SANTURKAR, D. TSIPRAS, L. ENGSTROM, B. TRAN and A. MADRY. 2019, «Adversarial examples are not bugs, they are features», in *Advances in Neural Information Processing Systems*, p. 125–136. 91

KRIZHEVSKY, A. 2009, «Learning multiple layers of features from tiny images», cahier de recherche. 105

LECUN, Y., C. CORTES and C. BURGES. 2010, «Mnist handwritten digit database», *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2. 105

LECUYER, M., V. ATLIDAKIS, R. GEAMBASU, D. HSU and S. JANA. 2019, «Certified robustness to adversarial examples with differential privacy», in *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, p. 656–672. 92

LI, B., C. CHEN, W. WANG and L. CARIN. 2019, «Certified adversarial robustness with additive noise», in *Advances in Neural Information Processing Systems*, p. 9464–9474. 92

LOSHCHILOV, I. and F. HUTTER. 2017, «Sgdr: Stochastic gradient descent with warm restarts», *International Conference on Learning Representations*. 106

MADRY, A., A. MAKELOV, L. SCHMIDT, D. TSIPRAS and A. VLADU. 2018, «Towards deep learning models resistant to adversarial attacks», in *International Conference on Learning Representations*. 91, 93, 94, 105, 106

MIRMAN, M., T. GEHR and M. VECHEV. 2018, «Differentiable abstract interpretation for provably robust neural networks», in *International Conference on Machine Learning*, p. 3578–3586. 92

PAPERNOT, N., P. MCDANIEL, I. GOODFELLOW, S. JHA, Z. B. CELIK and A. SWAMI. 2017, «Practical black-box attacks against machine learning», in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, ACM, New York, NY, USA, ISBN 978-1-4503-4944-4, p. 506–519, doi: 10.1145/3052973.3053009. 91

PAPERNOT, N., P. D. MCDANIEL and I. J. GOODFELLOW. 2016, «Transferability in machine learning: from phenomena to black-box attacks using adversarial samples», *CoRR*, vol. abs/1605.07277. 92

RAGHUNATHAN, A., J. STEINHARDT and P. LIANG. 2018, «Certified defenses against adversarial examples», in *International Conference on Learning Representations*. 92

RICE, L., E. WONG and Z. KOLTER. 2020, «Overfitting in adversarially robust deep learning», in *International Conference on Machine Learning*, PMLR, p. 8093–8104. 93, 105, 107

SZEGEDY, C., W. ZAREMBA, I. SUTSKEVER, J. BRUNA, D. ERHAN, I. GOODFELLOW and R. FERGUS. 2014, «Intriguing properties of neural networks», *International Conference on Learning Representations*. 91, 94

TORKAMANI, M. A. and D. LOWD. 2014, «On robustness and regularization of structural support vector machines», in *Proceedings of the 31st International Conference on Machine Learning, Proceedings of Machine Learning Research*, vol. 32, édité par E. P. Xing and T. Jebara, PMLR, Bejing, China, p. 577–585. 100

TORRALBA, A., R. FERGUS and W. T. FREEMAN. 2008, «80 million tiny images: A large data set for nonparametric object and scene recognition», *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, n° 11, p. 1958–1970. 105

VOROBEYCHIK, Y., M. KANTARCIOGLU, R. BRACHMAN, P. STONE and F. ROSSI. 2018, *Adversarial Machine Learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, ISBN 9781681733968. 91

WANG, Y., D. ZOU, J. YI, J. BAILEY, X. MA and Q. GU. 2019, «Improving adversarial robustness requires revisiting misclassified examples», in *International Conference on Learning Representations*. 93, 105

WONG, E. and J. Z. KOLTER. 2018, «Provable defenses against adversarial examples via the convex outer adversarial polytope», in *ICML, JMLR Workshop and Conference Proceedings*, vol. 80, JMLR.org, p. 5283–5292. 92

WONG, E., F. SCHMIDT, J. H. METZEN and J. Z. KOLTER. 2018, «Scaling provable adversarial defenses», in *Advances in Neural Information Processing Systems*, p. 8400–8409. 92

WU, D., S.-T. XIA and Y. WANG. 2020, «Adversarial weight perturbation helps robust generalization», *Advances in Neural Information Processing Systems*, vol. 33. 93, 105, 107

ZHANG, H., H. CHEN, C. XIAO, S. GOWAL, R. STANFORTH, B. LI, D. BONING and C.-J. HSIEH. 2020, «Towards stable and efficient training of verifiably robust neural networks», *International Conference on Learning Representation*. 92

ZHANG, H., Y. YU, J. JIAO, E. P. XING, L. E. GHAOUI and M. I. JORDAN. 2019, «Theoretically principled trade-off between robustness and accuracy», in *International Conference on Machine Learning*, p. 1–11. 15, 93, 95, 101, 104, 105, 106, 107, 108

ZHENG, Z. and P. HONG. 2018, «Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks», in *Neural Information Processing Systems*, p. 7924–7933. 91

Chapter 4

Robust Autoencoder-based State Estimation in Power Systems

Chapter 4 Abstract

In this chapter, we still address the first research question dedicated to increasing adversarial robustness. We present our second contribution, which falls the scope of smart grid systems. We follow the same idea as in Chapter 3, and we modify the training of the state estimator to ensure that it remains trustworthy under threats.

This chapter adapts the information-geometric regularizer previously introduced to build a robust state estimator in Smart Grid Systems. Due to their cyber component, such systems are highly vulnerable to threats. It is, therefore, crucial to enhancing their capability to face attacks. Indeed, through their interaction with citizens' lives, Smart Grid systems are critical systems.

In the following, we will present the state estimator problem in the linearized case, i.e., the DC model, and present our solution to build robust state estimators. It is worth noting that this work is among the first to rely on deep neural networks (*i.e.,* variational autoencoders in our case) in the field of robust state estimators for smart grid systems. Our extensive numerical experiments prove that our method is able to construct state estimators that are robust to state-of-the-art attacks in the case of multiple bus systems.

Contents

Abstract

Smart Grids are critical cyber-physical systems where monitoring is crucial, especially the process of state estimation. Since this task strongly depends on the reliability of power grid meters and their communication channels, it is vulnerable to cyber-attacks and, particularly, false data injection attacks (FDIAs), which are modifications on the meter readings that are often hard to detect. In this paper, we propose a method to construct a robust state estimator based on a variational autoencoder trained on the Fisher-Rao distance, which is a measure of dissimilarity between probability distributions. Then, we introduce a novel method to generate FDIAs that exploits knowledge of the state estimator and its learning procedure, for which we show effectiveness. Finally, numerical results and comparison with state-of-the-art methods confirm that our approach can archive similar estimation errors for clean and noisy (attacked) measurements.

4.1 Introduction

Smart Grids are complex systems composed of a physical layer, including generation, transmission, and distribution of electrical power, and a communication layer encompassing sensors and communication systems. Monitoring the network is essential to ensure self-healing, maintenance, interaction with the consumer while providing various target services [Abur and Exposito, 2004]. Consequently, power grid

This work was supported by the Natural Sciences and Engineering Research Council of Canada and McGill University in the framework of the NSERC/Hydro-Quebec Industrial Research Chair in Interactive Information Infrastructure for the Power Grid (IRCPJ406021-14).

systems surveillance requires strategic meters to measure different quantities, such as bus voltages, active or reactive power injections. These data are then processed at control centers, where general grid supervision is performed through accurate state estimations. This computational aspect of the Smart Grid management makes them vulnerable to unreliable data.

A *bad data* detector is usually implemented [MONTICELLI and GARCIA, 1983] to secure the system. However, it is not difficult to bypass this defense. For instance, attackers can alter the reliability of state estimation by constructing malicious (noisy) samples without being uncovered. The impact of attacks on power systems has been studied in SGOURAS and collab. [2014] and RICE and ALMAJALI [2014]. One potentially dangerous type of cyber-attacks that has gained attention over the last decade is known as False Data Injection Attacks (FDIAs) [LIU and collab., 2011]. FDIAs introduce attacks that are based on the actual power flow of the targeted grid. This knowledge allows the attacker to construct malicious manipulations of a subset of collected measurements across the grid. The main goal of an FDIA is, for instance, to disrupt the expected behavior or operation of the Smart Grids. In 2015, for example, a malicious attack was launched against a Ukrainian power plant. It deprived hundreds of thousands of houses of electricity for several hours [ALDERSON and DI PIETRO, 2016].

FDIAs can either be random, e.g., the goal is only to disrupt the state estimator, or targeted, e.g., the goal is to fool the estimator by inducing specific (target) values to the states. Stealth attacks [DÁN and SANDBERG, 2010; SUN and collab., 2019] are FDIAs designed so that their detection by the control center is made complex or even infeasible. Indeed, the issue of preventing cyber-attacks from disturbing the whole power system has recently been studied. Defense mechanisms are based on detecting FDIAs [ZONOUZ and collab., 2012], or introducing robustness against the loss of meters [ASHOK and collab., 2016]. The method investigated in HU and collab. [2017] estimates the error vectors to denoise the measurements before performing state estimation.

This paper introduces novel methods to generate attacks assuming full knowledge of both the state estimator and its learning procedure, and to defend against those powerful attacks. Our contributions are summarized as follow:

- We present a defensive scheme to prevent attacks from fooling the state estimator.
- We propose a method to create FDIAs using knowledge about the estimator's structure and learning procedure.
- Experimentally, we show that our attacks are highly efficient at disrupting state estimators.

- Finally, we design a robust estimator for various well-known electrical networks: IEEE 14/57/118-bus system and test it on state-of-the-art attack.

The rest of this article is organized as follows. In Section 4.2, we present an overview of the standard state estimation components, FDIAs, and MILP attack. We derive the defensive and attack schemes in Section 4.3. Experimental results are relegated to Section 4.4 where we prove the efficiency of the considered attack/defense mechanisms.

4.2 Background on State Estimation and Attacks

4.2.1 Bayesian approach for power state estimation

In general, the state is related to the measurement vector $\mathbf{Y} \in \mathbb{R}^m$ through the following nonlinear equation:

$$\mathbf{Y} = h(\mathbf{X}) + \mathbf{Z}, \tag{4.1}$$

where h is the measurement function and $\mathbf{Z} \in \mathbb{R}^m$ is additive noise. This is called the AC model in the literature [Abur and Exposito, 2004; Giannakis and collab., 2013].

Usually, under certain assumptions about the grid and its operating point, the problem can be linearized as follows:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{Z}, \tag{4.2}$$

where $\mathbf{H} \in \mathbb{R}^{m \times n}$ is the linearized Jacobian measurement matrix. This is called the DC model, and we will adopt it for the remainder of the paper.

State estimators are crucial components for the functioning of Smart Grids systems, but they are known to be sensitive to faulty samples. Bad data detectors are usually implemented to overcome this issue. They are a-posteriori detectors based on the l_p-norm (where typically $p = 2$) of the residual $\|\mathbf{Y} - \mathbf{H}\hat{\mathbf{X}}\|_p$, where $\hat{\mathbf{X}}$ is the estimated state. The sample is detected as faulty when the l_p-norm of the residual is greater than a threshold τ. Nevertheless, as we discuss next, bad data detectors are insufficient to overcome the problem of well-designed attacks.

4.2.2 False data injection attack

One of the most critical types of cyber-attacks are FDIAs. In Liu and collab. [2011], the authors present various methods to generate such corrupted samples to bypass the bad data detector under the DC assumption. Let us consider $\mathbf{Y_A}$ to be the attacked sample. We consider that

$$\mathbf{Y_A} = \mathbf{Y} + \mathbf{A}, \tag{4.3}$$

where $\mathbf{A}$ is the attack vector. It has been showed that, not to be identified as faulty by the bad data detector, the attack should be a combination of the column vectors of $\mathbf{H}$, i.e., $\mathbf{A} = \mathbf{H}\boldsymbol{c}$, for all $\boldsymbol{c}$.

There exist different scenarios where the attacker can, for example, only tamper with a certain number of measurements either of its choosing or imposed by the system. The attack can also be random or targeted. All these scenarios impose specific values for the vector $\boldsymbol{c}$. Intuitively, knowledge about the state estimator should give more power to the attacker. In the following, we present an attack method that uses that knowledge.

4.2.3 Review of the MILP Attack

The authors of KHEZRIMOTLAGH and collab. [2019] proposed an algorithm to create an undetected attack using the knowledge about the state estimator.

Their algorithm is composed of two parts. The first minimization problem allows the attacker to select which lines to overflow under the constraints that the generators' active powers are not modified, and the power flow behavior is respected.

The second minimization problem tries to minimize the attack vector's norm under the condition that perturbed lines (selected according to the resolution of the first minimization problem) are overflowed, meaning that their power flows are at most 20% above their maximum values. The attack vector should respect some conditions in order not to be detected. The power flow equations should be respected, and each component of the attack should not be too different from the value of the original observation. Finally, the total norm of the attack vector must be zero to satisfy the condition of zero-sum of all active power in the power system is met.

It is pretty easy to generate FDIAs that disrupt state estimation. Since state estimation is critical to ensure the proper functioning of Smart Grid systems, it is essential to ensure the reliability of the power system state estimator. Given that attacks are designed to remain undetected by the bad data detector, it appears to be fundamentally essential to improve the robustness of state estimators against all kinds of attacks, especially the most powerful ones. The following section aims at further developing this approach.

4.3 Defense Against False Data Injection Attacks

4.3.1 A Variational Autoencoder-based state estimator

We introduce a Variational Autoencoder (VAE) [KINGMA and WELLING, 2014] to build a robust state estimator. Our network is expected to solve Equation 4.2 by learning

$$\hat{\mathbf{X}} \sim q_{\boldsymbol{\phi}}(\mathbf{X}|\mathbf{Y}) \tag{4.4}$$

from the measurements $\mathbf{Y}$ in an unsupervised way, where $\boldsymbol{\phi}$ are the parameters of the encoder, and $\hat{\mathbf{Y}}$ is built from the estimate $\hat{\mathbf{X}}$ using the physical knowledge of $\mathbf{H}$ and of the noise $\mathbf{Z} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\mathbf{ZZ}})$:

$$\hat{\mathbf{Y}} \sim p(\hat{\mathbf{Y}}|\hat{\mathbf{X}}) = \mathcal{N}(\hat{\mathbf{Y}}; \mathbf{H}\hat{\mathbf{X}}, \boldsymbol{\Sigma}_{ZZ}). \tag{4.5}$$

The conditional pdf of $\mathbf{X}|\mathbf{Y}$ will be modeled as Gaussian,

$$q_{\boldsymbol{\phi}}(\mathbf{X}|\mathbf{Y}) = \mathcal{N}(\mathbf{X}; \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{Y}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{Y})), \tag{4.6}$$

where the mappings $\boldsymbol{\mu}_{\boldsymbol{\phi}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$ are learnt through a deep neural network encoder.

To increase the robustness of a state estimator against FDIAs, we need to ensure that the estimator makes similar predictions for clean and attacked samples. To this end, we build on our recent work [PICOT and collab., 2021] using tools from information geometry to introduce a training regularizer based on Fisher-Rao distance (FRD) [ATKINSON and MITCHELL, 1981] from which our robust VAE-based estimator is derived.

4.3.2 A new metric improving robustness against attacks

To obtain an estimator that is robust to corrupted samples, $\boldsymbol{\mu}_{\boldsymbol{\phi}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$, the learned autoencoder (see Subsection 4.2.1) has to perform similar on clean and noisy (attacked) measurements. Therefore, the defender must pursue two different goals:

- Correctly reconstruct the measurement vector from the estimated states. The estimator minimizes a reconstruction loss $\mathscr{L}(\mathbf{Y}, \mathbf{H}f_{\boldsymbol{\phi}}(\mathbf{Y}))$, where $\hat{\mathbf{X}} = f_{\boldsymbol{\phi}}(\mathbf{Y})$ is the estimated state vector and $f_{\boldsymbol{\phi}}(\mathbf{Y})$ is the sampling of $q_{\boldsymbol{\phi}}(\cdot|\mathbf{Y})$.

- Provide similar results through the mappings $\boldsymbol{\mu}_{\boldsymbol{\phi}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$ for both clean and corrupted measurement vectors, respectively $\mathbf{Y}$ and $\mathbf{Y}_{\mathbf{A}}$, through the minimization of the distance between the natural and corrupted distributions.

As we mentioned, one natural distance to consider is the FRD [ATKINSON and MITCHELL, 1981], which captures the distance between pdfs over the underlying

statistical manifold. Consider an attack mechanism $p(\mathbf{Y_A}|\mathbf{Y})$. For a given measurement vector $\mathbf{Y}$ and a sample of attacked measurements $\mathbf{Y_A}$, we can compare the pdfs $q_{\boldsymbol{\phi}}(\mathbf{X}|\mathbf{Y})$ and $q_{\boldsymbol{\phi}}(\mathbf{X}|\mathbf{Y_A})$.

If we assume that the problem is uni-dimensional and that $q_{\boldsymbol{\phi}}(\mathrm{X}|\mathrm{Y}) \sim \mathcal{N}(\mu, \sigma)$ and $q_{\boldsymbol{\phi}}(\mathrm{X}|\mathrm{Y_A}) \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma})$, the Fisher-Rao distance can be written as follows [ATKINSON and MITCHELL, 1981]:

$$d_{\mathrm{F}}(q_{\boldsymbol{\phi}}(\cdot|\mathrm{Y}), q_{\boldsymbol{\phi}}(\cdot|\mathrm{Y_A})) = \sqrt{2}\log \frac{\|(\frac{\mu}{\sqrt{2}}, \sigma) - (\frac{\tilde{\mu}}{\sqrt{2}}, -\tilde{\sigma})\| + \|(\frac{\mu}{\sqrt{2}}, \sigma_i) - (\frac{\tilde{\mu}}{\sqrt{2}}, \tilde{\sigma})\|}{\|(\frac{\mu}{\sqrt{2}}, \sigma) - (\frac{\tilde{\mu}}{\sqrt{2}}, -\tilde{\sigma})\| - \|(\frac{\mu}{\sqrt{2}}, \sigma_i) - (\frac{\tilde{\mu}}{\sqrt{2}}, \tilde{\sigma})\|}, \tag{4.7}$$

where μ and σ are the mean and variance of the clean estimated states, respectively, and $\tilde{\mu}$ and $\tilde{\sigma}$ are the corresponding attacked states ones.

For any dimension $d \geq 2$, if we assume that $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$ is diagonal, the FRD can be written as follows:

$$d_{\mathrm{R}}\big(q_{\boldsymbol{\phi}}(\cdot|\mathbf{Y}), q_{\boldsymbol{\phi}}(\cdot|\mathbf{Y_A})\big) = \sqrt{\sum_{\forall i} \big(d_{\mathrm{F}}(q^i_{\boldsymbol{\phi}}(\cdot|\mathbf{Y}), q^i_{\boldsymbol{\phi}}(\cdot|\mathbf{Y_A})\big)^2}, \tag{4.8}$$

where $q^i_{\boldsymbol{\phi}}$ indicates the i-th component of $q_{\boldsymbol{\phi}}$.

The defender's objective then becomes:

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \mathbb{E}[\mathscr{L}(\mathbf{Y}, \mathbf{H} f_{\boldsymbol{\phi}}(\mathbf{Y})] + \beta\, \mathbb{E}\left[d_{\mathrm{R}}(q_{\boldsymbol{\phi}}(\cdot|\mathbf{Y}), q_{\boldsymbol{\phi}}(\cdot|\mathbf{Y_A}))\right], \tag{4.9}$$

where $\mathscr{L}(\cdot,\cdot)$ is the reconstruction loss, and β is a hyperparameter controlling the trade-off between clean and attacked performances. It remains to define a defensive scheme using the above objective to learn a robust state estimator.

In order to train a state estimator as robust as possible, being able to generate the most powerful attacks is extremely important. The best possible method is to give the attacker full knowledge about the underlying training procedure.

4.3.3 Novel attacks exploiting the state-estimator knowledge

Knowledge about the state estimator is required to generate harmful attacks. In this setting, where the attackers have access to the state-estimator function and parameters $\boldsymbol{\phi}$, the attacker wants to maximize the expected error between the clean (non-corrupted) estimated state $\hat{\mathbf{X}}$ and the corrupted estimated state $\hat{\mathbf{X}}_\mathbf{A}$, i.e. $\mathbb{E}[\ell(\hat{\mathbf{X}}, \hat{\mathbf{X}}_\mathbf{A})]$. While at the same time, the attacker must guarantee that the bad data detector will not detect the corrupted samples (based on the generated noise), i.e., the attackers should minimize $\|\mathbf{Y_A} - \mathbf{H}\hat{\mathbf{X}}_\mathbf{A}\|_p$. Hence, the attacker's objective can be written as follows:

$$\mathbf{Y}^*_\mathbf{A} = \arg\max_{\mathbf{Y_A}} \ \ell(f_{\boldsymbol{\phi}}(\mathbf{Y}), f_{\boldsymbol{\phi}}(\mathbf{Y_A})) - \lambda \|\mathbf{Y_A} - \mathbf{H} f_{\boldsymbol{\phi}}(\mathbf{Y_A})\|_p, \tag{4.10}$$

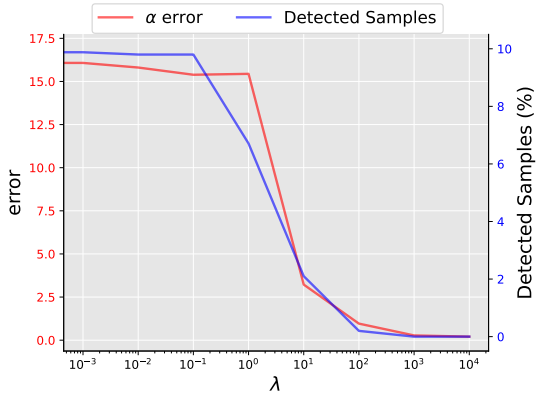


Figure 4.1: Error between natural and noisy samples using the state estimator-aware attack. ©2022 IEEE.

where $f_{\boldsymbol{\phi}}(\cdot)$ is the state estimator and λ is an hyperparameter controlling the trade-off between the power of the attack and the detection of this attack. In order to build such attack, we can use an arbitrary fidelity measure $\ell(\cdot,\cdot)$, e.g. an l_p-norm.

In order to create the attacks that will be the most harmful to a state estimator that defends according to Equation 4.9, the attacker should choose the Fisher-Rao distance between clean and attacked probability distributions as its fidelity measure.

4.3.4 General defensive framework

Algorithm 1 Review of the PGD Algorithm

INPUT: Define the step δ and the number of iterations n
INPUT: $\mathbf{Y}$
  $\mathbf{Y_A} \leftarrow \mathbf{Y}$
  for $i = 1..n$ **do**
    $\mathbf{Y_A} \leftarrow \mathbf{Y_A} + \delta\,\mathrm{sgn}(\nabla_{\mathbf{Y_A}}(g(\mathbf{Y},\mathbf{Y_A})))$, where
    $g(\mathbf{Y},\mathbf{Y_A}) = \mathscr{L}(f_{\boldsymbol{\phi}}(\mathbf{Y}), f_{\boldsymbol{\phi}}(\mathbf{Y_A})) - \lambda\|\mathbf{Y_A} - \mathbf{H}f_{\boldsymbol{\phi}}(\mathbf{Y_A})\|_p$ and sgn is the sign function.
  end for
OUTPUT: $\mathbf{Y_A}$

Given that the parameters $\boldsymbol{\phi}$ of a state estimator evolve during training, it is theoretically possible to attack the network at each training step. Therefore, the defender should generate attacked samples at each training step and then defend against them to ensure robustness. Two different steps are needed to train a robust state estimator:

- First, corrupted samples are generated using the proposed attack method in Equation 4.10. Since the underlying maximization problem Equation 4.10 is untractable, we generate noisy samples using the Projected Gradient Descent method (PGD) [MADRY and collab., 2018] (for further details, see Algorithm 1).

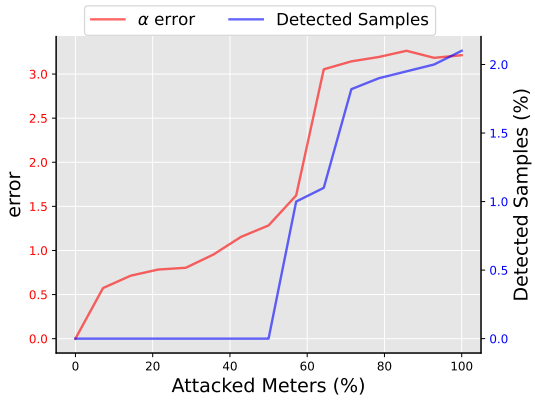


Figure 4.2: Errors between natural and corrupted performances as a function of the percentage of perturbed meters, for $\lambda = 10$ and without defense. ©2022 IEEE.

- Then, the defender updates its parameters $\boldsymbol{\phi}$ by approximately solving Equation 4.9.

From this methodology, we derive an approximate solution to a min-max problem which yields our robust estimator.

4.4 Numerical Results

In this section, we assess our proposed attack method's strength and evaluate the robust estimator's effectiveness on both our proposed attack and the MILP attack defined in KHEZRIMOTLAGH and collab. [2019].

4.4.1 Experimental set-up

Power network: We begin by considering the IEEE 14 bus system. The state vector is composed of the bus angles $\boldsymbol{\alpha}$. Later we present results on bigger IEEE bus systems (i.e., IEEE 57/118 bus systems), typically used as benchmarks. For each dataset, we define a training set (8000 samples used to train the model), a validation set (2000 samples used for the hyperparameters choices), and a testing set (1000 samples used for testing the resulting model).

Considered Meters: We use the active power P at each bus as the meters.

Autoencoder model:

- Since the DC model is linear, we employ a linear neural network to build the encoder, composed of a single layer using the measurement and the state size as input and output sizes.

- We use a physical knowledge-based decoder, defined by the *Kirchhoff* and *Ohm* laws.

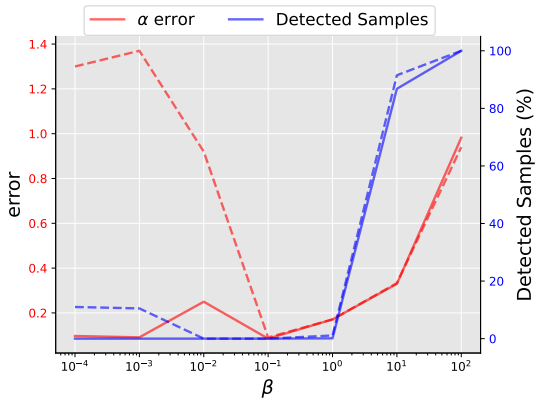


Figure 4.3: Estimated-angles mean errors and portions of detected samples for a robust estimator as a function of the β parameter. The natural errors are in plain lines, while the attacked ones are in dashes. ©2022 IEEE.

- To train the robust VAE model, we need to choose the fidelity measure $\ell(\cdot,\cdot)$ in Equation 4.9 and the reconstruction loss $\mathscr{L}(\cdot,\cdot)$ in Equation 4.10. For the latter, we have to ensure that the chosen network–along with its loss–can predict the states accurately. Hence, we chose $\mathscr{L}(\cdot,\cdot)$ as the l_1-norm between the ground-true measurement vector and the estimated one since it gives us the best state estimation results. Consequently, we also consider the l_1-norm for the bad data detector and the attacker. Then, we choose the FRD (see Equation 4.8) as the fidelity measure $\ell(\cdot,\cdot)$ to ensure that the attack/defense scheme is as balanced as possible.

Optimization set-up. We train for 200 epochs, using the Adam optimizer with a learning rate of 10^{-4} for the first 3/4 of learning and 10^{-5} afterward, and a weight decay of 5.10^{-4}.

Generation of the attack. We apply the formula in Equation 4.10, with the PGD method described in Algorithm 1. We set the step parameter δ to 0.03 and the number of iterations n to 10. In the final simulation, we consider the MILP attack presented in Subsection 4.2.3 which also has access to the knowledge of the state estimator.

4.4.2 Performance of the proposed attack

We aim at assessing the strength of the attack mechanism. To this end, we train a defenseless state estimator for the IEEE 14 bus system, where the mean angle clean error is equal to 0.0867 degrees. We generate the noisy (attacked) samples using the method presented in Subsection 4.3.3, and report the attack-induced mean angle error and the percentage of detected samples in Figure 4.1. Please observe that the impact of the λ hyperparameter (Equation 4.10) that controls the trade-off between the power of the attack and its detection is apparent in Figure 4.1. The bigger λ is,

the smaller attack-induced mean error and the number of detected samples are. For $\lambda = 100$, the number of detected samples is equal to 0. At the same time, the state estimation error caused by the attack based on the state estimator equals 1.08. Experimentally, it confirms that maximizing the Fisher-Rao distance between the original and the corrupted state attack is effective and can be undetected.

4.4.3 Influence of the number of attacked meters

It is reasonable to assume that the attacker often can only affect a subset among all meters in real-world scenarios. We, therefore, propose investigating the influence of the portion of attacked meters in the measurement vector. Once again, the attack-induced mean angle error and the percentage of detected samples are reported in Figure 4.2. We set $\lambda = 10$ to have a powerful attack and highlight the number of meter's influence on the power and the detection of the attack. We select the meters so that only the c most influential meters are kept, i.e., the l_0 of the attack vector is below c, with $c \in [0, 1]$.

As expected, the larger the portion of meters the attacker can affect, the more degrees of freedom it has, and thus, the more powerful the attack becomes. Nevertheless, the percentage of detected samples increases as well. Since a detected attack cannot be considered successful, the best trade-off between attack detection and power seems to be around 50% of corrupted meters. In the remaining, we assume that the attacks use $\lambda = 10$, and the attacker will affect 50% of the meters for each measurement vector.

Now, we focus on the defensive aspect of the state estimator.

4.4.4 Defensive performances

On the 14-bus net

We train a robust state estimator, using the method detailed in Subsection 4.3.4. First, we need to study the influence of the hyperparameter β that controls the trade-off between natural and attacked performances. The mean angle errors and the percentage of noisy detected samples are reported in Figure 4.3.

The trade-off between clean and attacked performances is visible. While increasing the β hyperparameter provides better performances on attacked samples, it also worsens the clean ones. For $\beta \geq 1$, the VAE starts to give more importance to estimating the attacked and clean natural samples in the same way and no longer tries to predict the clean measurements correctly. The best trade-off between clean and attacked performances was found for $\beta = 0.1$.

Table 4.1: Clean and Attacked Mean Error for Defenseless and Robust Estimator for different Bus Systems. ©2022 IEEE.

Bus System		14buses	57buses	118buses
Defenseless Estimator	Clean Error	0.0867	0.1171	0.0969
	Attacked Error	1.37	7.4910	2.1568
	MILP Attack	0.4412	0.1248	0.1149
Robust Estimator	Clean Error	0.0989	0.1527	0.1028
	Attacked Error	0.1021	0.1535	0.1020
	MILP Attack	0.0982	0.1531	0.0994

On other classical power networks

We choose to follow the classical IEEE bus systems with three different numbers of buses, namely: small (14 buses), medium (57 buses), and large (118 buses). We train a defenseless and a robust estimator for $\lambda = 10$ and 50% of corrupted meters. The optimal β value is 0.1 for all cases. We then test both estimators on clean observations, attacked observation using our proposed method, and attacked observation using the MILP attack [KHEZRIMOTLAGH and collab., 2019].

We report the mean absolute state error on clean and attacked observations for both a defenseless and a robust state estimator for all considered power networks in Table 4.1. The clean and attacked errors are pretty similar for all networks and for both considered attacks (our attack and the MILP attack) when a robust estimator is trained, while the amount of natural performances lost by the robust training is between 0 and 0.04 degrees. Therefore, it is possible to construct a robust state estimator that improves robustness against state estimator-aware attacks while not decreasing the performance significantly on clean samples.

4.5 Conclusion

In this paper, we introduced a novel method to generate powerful stealth false data injection attacks based on the structure and training procedure of the state estimator that the attacker aims at fooling. We have experimentally proven that this new generation of attacks introduces state estimation error.

We also introduced a method to learn a robust state estimator using a geometric information distance, known as the Fisher-Rao distance, based on a min-max game between the estimator and the attacker. On multiple benchmarks of power networks, we have experimentally proven that it is possible to train a state estimator that improves its robustness against attacks while not decreasing the performance significantly on clean samples.

Finally, it is worth mentioning that our results are based on the DC model assumption. However, we plan to generalize the proposed method to the more general

AC model setup.

Chapter 4 Conclusion

In this chapter, we addressed the problem of leveraging the knowledge about the output space to increase a state estimator's robustness. We modified our regularizer-based method on the Fisher-Rao distance to apply it to smart grid systems.
We are among the first to use deep models to train a state estimator, based on Variational AutoEncoders, that estimates the appropriate state even under threat. In this work, we restricted ourselves to the linearized case. However, this hypothesis is not necessarily realistic for every bus system. In the next chapter, we will extend our work to be applicable to non-linear state estimators.

4.6 References

ABUR, A. and A. G. EXPOSITO. 2004, *Power system state estimation: theory and implementation*, CRC press. 120, 122

ALDERSON, D. and R. DI PIETRO. 2016, «Operational technology: Are you vulnerable?», *Governance Directions*, vol. 68, nº 6, p. 339–343. 121

ASHOK, A., M. GOVINDARASU and V. AJJARAPU. 2016, «Attack-resilient measurement design methodology for state estimation to increase robustness against cyber attacks», in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, IEEE, p. 1–5. 121

ATKINSON, C. and A. F. MITCHELL. 1981, «Rao's distance measure», *Sankhyā: The Indian Journal of Statistics, Series A*, p. 345–365. 124, 125

DÁN, G. and H. SANDBERG. 2010, «Stealth attacks and protection schemes for state

estimators in power systems», in *2010 first IEEE international conference on smart grid communications*, IEEE, p. 214–219. 121

GIANNAKIS, G. B., V. KEKATOS, N. GATSIS, S.-J. KIM, H. ZHU and B. F. WOLLENBERG. 2013, «Monitoring and optimization for power grids: A signal processing perspective», *IEEE Signal Processing Magazine*, vol. 30, nº 5, p. 107–128. 122

HU, Q., D. FOOLADIVANDA, Y. H. CHANG and C. J. TOMLIN. 2017, «Secure state estimation and control for cyber security of the nonlinear power systems», *IEEE Transactions on Control of Network Systems*, vol. 5, nº 3, p. 1310–1321. 121

KHEZRIMOTLAGH, D., J. KHAZAEI and A. ASRARI. 2019, «Milp modeling of targeted false load data injection cyberattacks to overflow transmission lines in smart grids», in *2019 North American Power Symposium (NAPS)*, IEEE, p. 1–7. 123, 127, 130

KINGMA, D. P. and M. WELLING. 2014, «Auto-encoding variational bayes», *International Conference on Learning Representations*. 124

LIU, Y., P. NING and M. K. REITER. 2011, «False data injection attacks against state estimation in electric power grids», *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, nº 1, p. 1–33. 121, 122

MADRY, A., A. MAKELOV, L. SCHMIDT, D. TSIPRAS and A. VLADU. 2018, «Towards deep learning models resistant to adversarial attacks», in *International Conference on Learning Representations*. 126

MONTICELLI, A. and A. GARCIA. 1983, «Reliable bad data processing for real-time state estimation», *IEEE Transactions on Power Apparatus and Systems*, , nº 5, p. 1126–1139. 121

PICOT, M., F. MESSINA, M. BOUDIAF, F. LABEAU, I. B. AYED and P. PIANTANIDA. 2021, «Adversarial robustness via fisher-rao regularization», *CoRR*, vol. abs/2106.06685. 124

RICE, E. B. and A. ALMAJALI. 2014, «Mitigating the risk of cyber attack on smart grid systems», *Procedia Computer Science*, vol. 28, p. 575–582. 121

SGOURAS, K. I., A. D. BIRDA and D. P. LABRIDIS. 2014, «Cyber attack impact on critical smart grid infrastructures», in *ISGT 2014*, IEEE, p. 1–5. 121

SUN, K., I. ESNAOLA, S. M. PERLAZA and H. V. POOR. 2019, «Stealth attacks on the smart grid», *IEEE Transactions on Smart Grid*, vol. 11, nº 2, p. 1276–1285. 121

ZONOUZ, S., K. M. ROGERS, R. BERTHIER, R. B. BOBBA, W. H. SANDERS and T. J. OVERBYE. 2012, «Scpse: Security-oriented cyber-physical state estimation for power grid

critical infrastructures», *IEEE Transactions on Smart Grid*, vol. 3, nº 4, p. 1790–1799. 121

Chapter 5

Robust State Estimation Against Adversarial Noise

Chapter 5 Abstract

This chapter represents the final part of our answer to the first research question dedicated to increasing adversarial robustness. We present our third and final contribution. In the previous chapter, we present the extension of the framework presented in Chapter 3 to the state estimator's problem in smart grid systems. To build our robust state estimator in Chapter 4, we made the DC model assumption, i.e., that the state estimation problem was linear.
However, this linear assumption does not always hold.
In the following, we will present the state estimator problem in the non-linearized case, i.e., the AC model, which is the realistic case since all electrical systems are actually always non-linear. We present our solution to build robust non-linear state estimators. Our extensive numerical experiments prove that our method is able to construct state estimators that are robust to state-of-the-art attacks in the case of multiple bus systems.

Abstract

Smart Grids are critical systems where state estimation is critical due to monitoring purposes. Through this computational aspect, Smart Grids are vulnerable to cyber-attacks, and in particular, to False Data Injection Attacks (FDIAs). In this paper, we begin by introducing two novel methods to generate FDIAs that exploit knowledge of the state estimator: a deterministic process, creating undetectable attack for each observation vector, and a random process, learning a probability distribution depending on the observations. Then, we present a method to construct a robust state estimator based on a variational autoencoder trained using ideas from adversarial deep learning to improve its robustness capabilities. Finally, we test our robust estimator for three different

power systems: the IEEE 14/57/118 buses systems. The results show that our approach can provide similar estimation errors for clean and noisy (attacked) measurements, also showing that the estimator is not sensitive to the attacker strategy.

5.1 Introduction

5.1.1 Motivation

Smart Grids are cyber-physical electrical systems where monitoring is an essential task. Real-time analysis of the network is made possible through the strategic deployment of meters on the grid. These meters are then processed to assess the state of the grid through accurate state estimation. This computational aspect, while allowing self-healing, maintenance, and interaction with the consumer while providing various services [ABUR and EXPOSITO, 2004], also increases the vulnerability of smart grids to unreliable data and malicious agents.

To prevent unreliable data from disrupting the normal operation of a smart grid, *bad data* detectors are usually implemented and included in the state estimation process. However, it is well-known that malicious agents can bypass this protection using well-crafted attacks and alter the reliability of state estimation without being uncovered. In these cyber-attacks, known as False Data Injection Attacks (FDIAs) [LIU and collab., 2011], the attacker makes use of the knowledge about the power grid and the monitoring system to craft the attack. FDIAs attempt to disrupt the state estimation process through the manipulation of specific measurements. The attacks can either be random, e.g., the goal is only to disrupt the state estimator, or targeted, e.g., the goal is to induce specific (targeted) values to the states. Stealth attacks [DÁN and SANDBERG, 2010; SUN and collab., 2019] are FDIAs designed so that their detection by the control center is made complex or even infeasible. The problem of defense against FDIAs is of critical importance in practice. For instance, in 2015, a malicious attack was successfully launched against a Ukrainian power plant. This deprived hundreds of thousands of houses of electricity for several hours [ALDERSON and DI PIETRO, 2016].

5.1.2 Related work

FDIAs have been widely studied since their introduction by LIU and collab. [2011]. Power system vulnerability studies have been conducted by developing strategies to efficiently design attack schemes [HUG and GIAMPAPA, 2012; KOSUT and collab., 2010; LIU and collab., 2011; RAHMAN and MOHSENIAN-RAD, 2013; YUAN and collab., 2011]. The vulnerability has been defined as the minimum number of compromised meters

required to successfully disrupt the state estimation [SANDBERG and collab., 2010]. Unfortunately this is a complex cardinality minimization problem which is hard to solve, particularly for large grids with several meters installed. In fact, the authors in HENDRICKX and collab. [2014] showed that this problem is an NP-hard problem and presented a relaxation hypothesis to solve it efficiently. A similar approach has been considered by the authors in SOU and collab. [2013]. The impact caused by FDIAs has been studied from different perspectives such as the energy deceiving aspect [LIANG and collab., 2016] or the economic losses [LIANG and collab., 2016; TAJER, 2017; XIE and collab., 2010].

Most of the research has been done according to the DC model assumption [DÁN and SANDBERG, 2010; KOSUT and collab., 2010; LIU and collab., 2011; PASQUALETTI and collab., 2011; SANDBERG and collab., 2010; XIE and collab., 2010; YUAN and collab., 2011], which leads to a simple linear model for the measurements as a function of the state. These works show precisely how FDIAs can bypass bad data detectors. A comprehensive review of the security problem under the DC-model assumption can be found in LIANG and collab. [2016]. A few works have focused on AC-based FDIAs [HUG and GIAMPAPA, 2012; JIN and collab., 2018; KEKATOS and collab., 2017; LIANG and collab., 2014; TEIXEIRA and collab., 2015; ZHU and GIANNAKIS, 2012], which considers the nonlinear relation between the state and the measurements. On the other hand, joint cyber and physical attacks have also been studied [SOLTAN and collab., 2016], leading to the study of detection and recovery of information from line failure under DC and AC assumption [SOLTAN and collab., 2018; SOLTAN and ZUSSMAN, 2018].

Three main types of defense strategies have been developed in the literature. The first one is the detection of the FDIAs [KOSUT and collab., 2010; PASQUALETTI and collab., 2011], which can be used to discarding the compromised measurements. The second one is based on encryption, authentication and key management [DÁN and SANDBERG, 2010; TEIXEIRA and collab., 2015; WANG and LU, 2013] in order to protect the communication channel between the meters and the control center. Finally, Robust State Estimation (RSE) under the DC assumption has been created to develop state estimations that are robust against bad data [CELIK and ABUR, 1992; KOTIUGA and VIDYASAGAR, 1982; MILI and collab., 1994; ZHU and GIANNAKIS, 2012]. An AC-based solution to the RSE problem has been investigated by ZHANG and collab. [2017], but it requires many relaxation assumptions. It should be noted that these defense strategies are complementary to each other and can be used simultaneously in practice. In this work, we focus on the latter approach to protect the grid against FDIAs.

5.1.3 Contributions

In this paper, we propose a novel method to craft robust state estimators based on a game between the estimator and a simulated attacker using a deep learning approach. The main idea is that if the simulated attacker and the estimator are trained jointly, the estimator will become robust in the process. Our idea is inspired by the field of adversarial machine learning [VOROBEYCHIK and KANTARCIOGLU, 2018].

Our contributions can be summarized as follows:

- We present a scheme to train a robust state estimator to prevent attacks from disrupting the state estimator based on an adversarial training approach.
- We propose two frameworks to generate attacks: a deterministic framework, where the attack vector is constructed using the gradients of the loss function with respect to the input, and a random gaussian framework, where the mean and the standard variation both depend on the measurements. The latter attack can be seen as a generalization of the gaussian attacks presented in SUN and collab. [2019].
- We compare the deterministic and random attacks on the IEEE 14 bus system and find that random attacks seem better suited to train a robust state estimator.
- Finally, we test our defense scheme against several attacks on the IEEE 14/57/118 bus systems and compare its performance with a state-of-the-art method. We find that our approach yields a more robust estimator.

The rest of the paper is organized as follows. In Section 5.2, we present a review of the state estimation problem, followed by an overview of false data injection attacks. In particular, we review a state-of-the-art defense mechanism and an attack method to later use as our benchmark. In Section 5.3, we present our defensive framework along with the proposed attacks. Finally, in Section 5.4, we experimentally show the effectiveness of our robust state estimator.

5.2 Background on State Estimation and Attacks

5.2.1 Bayesian approach for power state estimation

Let $\mathbf{x} = |\mathbf{x}|e^{i\arg\mathbf{x}} \in \mathbb{C}^n$ be the state vector (latent variables) of the power grid, assumed to be random with a prior probability density function (pdf) $p(\mathbf{x})$, and $\mathbf{y} = |\mathbf{y}|e^{i\arg\mathbf{y}} \in \mathbb{C}^m$ the measurement vector. In general, these are related through the following nonlinear equation:

$$\mathbf{y} = \boldsymbol{h}(\mathbf{x}) + \mathbf{z}, \tag{5.1}$$

where $\boldsymbol{h}$ is the measurement function (dependent on the grid topology, line impedances, etc.) and $\mathbf{z} \in \mathbb{C}^m$ is additive noise. This is called the AC model in the literature [ABUR and EXPOSITO, 2004; GIANNAKIS and collab., 2013].

We define a (parametric) state estimator as a function $f_{\boldsymbol{\phi}} : \mathbb{C}^m \to \mathbb{C}^n$ such that $\hat{\mathbf{x}} = f_{\boldsymbol{\phi}}(\mathbf{y})$. The parameters $\boldsymbol{\phi}$ can be obtained by minimizing a risk function (see Section 5.3).

State estimators are crucial components for the functioning of smart grid systems, but they are known to be sensitive to faulty samples. Bad data detectors are usually implemented to overcome this issue. They are a-posteriori detectors based on the l_p-norm (where typically $p = 2$) of the residual vector

$$\mathbf{r} = \mathbf{y} - \boldsymbol{h}(\hat{\mathbf{x}}). \tag{5.2}$$

Concretely, the sample is detected as faulty when the $\|\mathbf{r}\|_p > \tau$, where τ is a threshold chosen appropriately to control the trade-off between missed and false detections. Nevertheless, bad data detectors are insufficient to overcome the problem of well-designed attacks. In particular, a theoretical bound on the maximal number of meters that one can attack in order to remain undetected is derived in JIN and collab. [2018].

5.2.2 False Data Injection Attacks

In order to craft an FDIA under the AC model (Equation 5.1), the attacker can solve the following optimization problem [JIN and collab., 2018]:

$$\begin{aligned} \max_{\mathbf{x_a}, \mathbf{a}} \quad & g(\mathbf{x_a}) \\ \text{s.t.} \quad & \boldsymbol{h}(\mathbf{x_a}) + \mathbf{z} = \mathbf{y} + \mathbf{a} = \mathbf{y_a} \\ & \|\mathbf{a}\|_0 \leq c, \end{aligned} \tag{5.3}$$

where $\mathbf{x_a}$ is the corrupted state, $\mathbf{a}$ is the attack vector, $\mathbf{y_a} = \mathbf{y} + \mathbf{a}$ is the attacked observation, and $g(\cdot)$ is the objective of the attacker. The condition on the l_0-norm of the attacked vector allows the attacker to bypass the bad data detector by attacking only a subset of sensors. Since the minimization problem in Equation 5.3 is widely non-convex, different relaxations are proposed in JIN and collab. [2018] to solve Equation 5.3.

Notice that the attacker objective can be different based on its specific goal, for example:

- Target state attack, where $g(\mathbf{x_a}) = \|\mathbf{x_a} - \mathbf{x}_{\text{target}}\|_2^2$, which will put the corrupted state to the targeted value $\mathbf{x}_{\text{target}}$.
- Voltage collapse attack, where $g(\mathbf{x_a}) = \|\mathbf{x_a}\|_2^2$, which will lead the estimator to believe that the voltage is low.

- State deviation attack, where $g(\mathbf{x_a}) = -\|\mathbf{x_a} - \hat{\mathbf{x}}\|_2^2$, which will force the corrupted state to be different from the original predicted one.

The optimization problem in Equation 5.3 requires to solve precisely the state estimation problem for each attack vector $\mathbf{a}$ in order to obtain $\mathbf{x_a}$. If we assume that the attacker knows the structure of the state estimator f_ϕ, it is reasonable and convenient to replace $\mathbf{x_a}$ with $f_\phi(\mathbf{y}+\mathbf{a})$. This leads to the following problem for the attack design:

$$\begin{aligned} \max_{\mathbf{a}} \quad & g(f_\phi(\mathbf{y}+\mathbf{a})) \\ \text{s.t.} \quad & \|\mathbf{a}\|_0 \leq c, \end{aligned} \tag{5.4}$$

We focus on this formulation for the remaining of the paper.

A variety of attacks has recently been proposed to attack AC State Estimator. While some try to attack without any knowledge about the grid to attack, we focus on attacks with full knowledge about it. This can be considered as a worst-case scenario, which is a reasonable criterion to study the vulnerability of critical systems such as smart grids.

JIN and collab. [2017] presented a semi-definite relaxation to create a convex attack optimization to create powerful and sparse attacks. The sparsity constraint is relaxed as an l_1-norm constraint on the additive noise. While their method has been presented for targeted attacks, it can be easily extended to any of the three attacker's objectives presented in Subsection 5.2.2.

While crafting attacks that disrupt the state estimator's functioning is relatively easy, building efficient defenses against said attacks is a more challenging task.

5.2.3 Review of the LASSO robust state estimators

There exists in the literature a few works that focus on crafting robust state estimators [JIN and collab., 2019; ZHANG and collab., 2017]. We decided to focus on the LASSO solution proposed in JIN and collab. [2019]. First, they introduce a new basis for the state estimation problem, where the AC problem is, in fact, linear. On this new basis, the state is no longer directly optimized. Instead, the new state is a combination of the original one. In order to obtain the robust state estimator, two steps are needed. The first one is the optimization of a lasso regression, where the l_2 reconstruction error is regularized by the l_1 norm of the noise. The second step consists of the retrieval of the original state from the new basis one.

5.3 Proposed Robust State Estimator

5.3.1 A Variational AutoEncoder based state estimator

We propose to use a Variational AutoEncoder (VAE) [KINGMA and WELLING, 2014] to build a robust state estimator. Our network is expected to provide a state estimator by learning a conditional distribution $q_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{y})$ such that $\hat{\mathbf{x}} = f_{\boldsymbol{\phi}}(\mathbf{y}) \sim q_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{y})$ from the measurements $\mathbf{y}$ in an unsupervised way. Here, $\boldsymbol{\phi}$ are the parameters of the encoder. Notice that we can then reconstruct the estimate observation $\hat{\mathbf{y}}$ from the estimate $\hat{\mathbf{x}}$ using the physical knowledge of the grid (to obtain h) and of the noise distribution $\mathbf{z}$ (more specifically, its covariance matrix $\boldsymbol{\Sigma}_{\mathbf{z}}$). The equivalent minimization problem is:

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \mathrm{L}(\boldsymbol{\phi}), \tag{5.5}$$

where $\mathrm{L}(\boldsymbol{\phi})$ is the risk of the state estimator. To obtain an estimator that is robust to corrupted samples, we need to ensure similar performance for clean and attacked measurements. Therefore, the risk should be defined based on two goals:

1. Correctly estimate the states from the observations $\mathbf{y}$.

2. Provide similar results for both clean and corrupted measurement vectors, respectively $\mathbf{y}$ and $\mathbf{y_a}$.

To ensure both goals are met, we propose the following risk:

$$\mathrm{L}(\boldsymbol{\phi}) = \mathrm{L}_{\text{clean}}(\boldsymbol{\phi}) + \beta \mathrm{L}_{\text{attacked}}(\boldsymbol{\phi}), \tag{5.6}$$

where minimizing $\mathrm{L}_{\text{clean}}(\boldsymbol{\phi})$ will ensure that the clean states are rightfully estimated, while minimizing $\mathrm{L}_{\text{attacked}}(\boldsymbol{\phi})$ will ensure clean and attacked mapping are similar. β is a fixed hyperparameter controlling the trade-off between the goals.

To rightfully estimate the clean states, we consider:

$$\mathrm{L}_{\text{clean}}(\boldsymbol{\phi}) = \mathbb{E}[\mathscr{L}(\mathbf{y}, \boldsymbol{h}(\hat{\mathbf{x}})], \tag{5.7}$$

where $\mathscr{L}(\cdot,\cdot)$ is the reconstruction loss. In general, this optimization problem is highly non-convex and has an infinite of solutions. To force the state estimator to find the solution we expect, we impose the condition that the amplitude of the predicted state $\hat{\mathbf{x}}$ has to be within 10% of 1 p.u. The new risk therefore becomes:

$$\mathrm{L}_{\text{clean}}(\boldsymbol{\phi}) = \mathbb{E}\left[\mathscr{L}(\mathbf{y}, \boldsymbol{h}(\hat{\mathbf{x}})) + c \cdot ||1 - |\hat{\mathbf{x}}|\,||_1\right], \tag{5.8}$$

where

$$c = \begin{cases} 1, & \text{if } ||1 - |\hat{\mathbf{x}}|\,||_1 > 0.1 \\ 0, & \text{otherwise.} \end{cases} \tag{5.9}$$

The conditional distribution $p(\mathbf{x}|\mathbf{y})$ will be modeled as Gaussian, i.e., $q_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{y}), \boldsymbol{\Sigma}_{\boldsymbol{\phi}}(\mathbf{y}))$ where the mappings $\boldsymbol{\mu}_{\boldsymbol{\phi}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$ are learnt through a deep neural network encoder. Notice that the decoder is fixed (i.e., it is not required to learn it) since $p(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}; \boldsymbol{h}(\mathbf{x}), \boldsymbol{\Sigma}_{\mathbf{z}})$.

The question now is how we should choose $\mathrm{L}_{\text{attacked}}(\boldsymbol{\phi})$ to ensure that clean and attacked states are similar. To answer this question, we build on our recent work [PICOT and collab., 2021] using tools from information geometry to introduce a risk function based on Fisher-Rao distance (FRD) [ATKINSON and MITCHELL, 1981; PINELE and collab., 2020] between two distributions. We now develop this idea to define $\mathrm{L}_{\text{attacked}}(\boldsymbol{\phi})$ for our robust VAE-based state estimator.

5.3.2 A new metric to improve robustness against attacks

As previously mentioned, the choice of the risk to ensure that clean and attacked states are similarly estimated is important. The output of the variational autoencoder being a probability distribution, one natural distance to consider is the FRD [ATKINSON and MITCHELL, 1981; PINELE and collab., 2020], which captures the distance between distributions over the underlying statistical manifold (defined by the encoder neural network in this case).

Consider an attack mechanism that outputs an attack observation $\mathbf{y_a}$ given a clean one $\mathbf{y}$. We can compare the predicted distributions $q_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{y})$ and $q_{\boldsymbol{\phi}}(\mathbf{x}|\mathbf{y_a})$. If we first assume that the problem is one-dimensional and that $q_{\boldsymbol{\phi}}(x|y) = \mathcal{N}(x; \mu, \sigma^2)$ and $q_{\boldsymbol{\phi}}(x|y_a) = \mathcal{N}(x; \tilde{\mu}, \tilde{\sigma}^2)$, the Fisher-Rao distance can be written as follows [ATKINSON and MITCHELL, 1981]:

$$d_{\mathrm{F}}(q_{\boldsymbol{\phi}}(\cdot|y), q_{\boldsymbol{\phi}}(\cdot|y_a)) = \sqrt{2}\log \frac{\|(\frac{\mu}{\sqrt{2}}, \sigma) - (\frac{\tilde{\mu}}{\sqrt{2}}, -\tilde{\sigma})\| + \|(\frac{\mu}{\sqrt{2}}, \sigma) - (\frac{\tilde{\mu}}{\sqrt{2}}, \tilde{\sigma})\|}{\|(\frac{\mu}{\sqrt{2}}, \sigma) - (\frac{\tilde{\mu}}{\sqrt{2}}, -\tilde{\sigma})\| - \|(\frac{\mu}{\sqrt{2}}, \sigma) - (\frac{\tilde{\mu}}{\sqrt{2}}, \tilde{\sigma})\|}. \tag{5.10}$$

Now, if we assume that $\boldsymbol{\Sigma}_{\boldsymbol{\phi}}$ is diagonal for simplicity, the FRD can be written as follows:

$$d_{\mathrm{R}}\big(q_{\boldsymbol{\phi}}(\cdot|\mathbf{y}), q_{\boldsymbol{\phi}}(\cdot|\mathbf{y}_a)\big) = \sqrt{\sum_{i=1}^{n} \big(d_{\mathrm{F}}(q^i_{\boldsymbol{\phi}}(\cdot|\mathbf{y}), q^i_{\boldsymbol{\phi}}(\cdot|\mathbf{y_a})\big)^2}, \tag{5.11}$$

where $q^i_{\boldsymbol{\phi}}$ is the i-th component of $q_{\boldsymbol{\phi}}$.

A natural choice for $\mathrm{L}_{\text{attacked}(\boldsymbol{\phi})}$, using FRD, is:

$$\mathrm{L}_{\text{attacked}(\boldsymbol{\phi})} = \mathbb{E}[d_{\mathrm{R}}(q_{\boldsymbol{\phi}}(\cdot|\mathbf{y}), q_{\boldsymbol{\phi}}(\cdot|\mathbf{y_a})]. \tag{5.12}$$

The remaining question is how to generate powerful attacks $\mathbf{y_a}$ on which to train our robust state estimator. In the following, we will present two attack mechanisms that make use of the full knowledge about the targeted state estimator.

Algorithm 2 Deterministic attack generation (PGD Algorithm)

INPUT: Define the step δ and the number of iterations n_{PGD}
INPUT: $\mathbf{y}$
  $\mathbf{y_a} \leftarrow \mathbf{y} + \mathbf{a}$ with $\mathbf{a} \sim \text{Unif}([-0.01, 0.01])$
  for $i = 1..n_{\text{PGD}}$ **do**
    Compute $l(\mathbf{y}, \mathbf{y_a}; \boldsymbol{\phi})$ from Equation 5.13.
    $\mathbf{a} \leftarrow \delta \cdot \text{sgn}(\nabla_{\mathbf{y}}(l(\mathbf{y}, \mathbf{y_a}; \boldsymbol{\phi})))$, where sgn is the element-wise sign function.
    $\mathbf{y_a} \leftarrow \mathbf{y_a} + \mathbf{a}$
  end for
OUTPUT: $\mathbf{y_a}$

5.3.3 Attack mechanisms using full knowledge about the state estimator

The attacker's goal is to disrupt the regular operation of the state estimator. We assume that its goal is to maximize the expected error between the original predicted state values $\hat{\mathbf{x}}$ and the state estimation of the attacked observation $\hat{\mathbf{x}}_a$, corresponding to the 3rd case of the objectives in Subsection 5.2.2.

In the setting where full access to the state estimator to attack is available, the attacker wants to maximize, for each observation $\mathbf{y}$, the error between the clean (non-corrupted) estimated state $\hat{\mathbf{x}}$ and the corrupted estimated state $\hat{\mathbf{x}}_\mathbf{a}$, i.e. $\ell(\hat{\mathbf{x}}, \hat{\mathbf{x}}_\mathbf{a}; \boldsymbol{\phi})$, where $\ell(\cdot, \cdot; \cdot)$ is an error loss. At the same time, the attacker must guarantee that the generated sample will remain undetected by the bad data detector, i.e., the attacker should minimize $\|\mathbf{y_a} - \boldsymbol{h}(\hat{\mathbf{x}}_\mathbf{a})\|_p$. The attacker's objective can therefore be written as:

$$l(\mathbf{y}, \mathbf{y_a}; \boldsymbol{\phi}) = \ell(\hat{\mathbf{x}}, \hat{\mathbf{x}}_\mathbf{a}; \boldsymbol{\phi}) - \lambda \|\mathbf{y_a} - \boldsymbol{h}(\hat{\mathbf{x}}_\mathbf{a})\|_p, \tag{5.13}$$

where λ is a hyperparameter controlling the trade-off between the power of the attack and its capability to remain undetected.

The attacker's objective can be optimized in two ways, either in a deterministic way or following a random process. In the deterministic case, the attacker maximizes its objective for each observation, i.e., for a given observation $\mathbf{y}$,

$$\mathbf{y}_\mathbf{a}^* = \arg\max_{\mathbf{y_a}} l(\mathbf{y}, \mathbf{y_a}; \boldsymbol{\phi}). \tag{5.14}$$

The attacker can also generate random attacks following:

$$\mathbf{y}_\mathbf{a}^* \sim q^*(\mathbf{y}_\mathbf{a}|\mathbf{y}) = \underset{q(\mathbf{y}_\mathbf{a}|\mathbf{y}):\, \mathbf{y}_\mathbf{a}\sim q(\mathbf{y}_\mathbf{a}|\mathbf{y})}{\arg\max} \mathbb{E}[l(\mathbf{y}, \mathbf{y}_\mathbf{a}; \boldsymbol{\phi})], \tag{5.15}$$

It should be mentioned that this attacker's objective is similar to the one in SUN and collab. [2019] if we set ℓ to be the mutual information between the attacked observation and the state.

In the following, we will present two algorithms to generate deterministic and random attacks according to the optimization problems in Equation 5.14 and Equation 5.15, respectively.

Deterministic attack mechanism

Since the problem in Equation 5.14 is extremely non-convex and hence widely intractable, we consider a simple approach to approximate its solution. Concretely, we consider a Projected Gradient Descent (PGD) method to generate the attacked measurements. This method was introduced in the field of adversarial machine learning to generate adversarial examples for computer vision problems [MADRY and collab., 2018]. It is an iterative method based on the sign of the gradient of the objective to maximize. The procedure is explained in detail in Algorithm 2. Notice that each attack is independent of the others and the computation of the attack vector $\mathbf{a}$ must be done from scratch for each measurement vector $\mathbf{y}$.

Algorithm 3 Random attack generation (noisy channel optimization)

INPUT: Define the learning rate α_c and the number of steps n_c
INPUT: input data $\{\mathbf{y}^1, .., \mathbf{y}^m\}$
  Initialize the parameters $\boldsymbol{\theta}$, and fix $\boldsymbol{\phi}$
  $n = 0$
  while $n < n_c$ **do**
    Draw a minibatch $\{\mathbf{y}^{\pi_1}, .., \mathbf{y}^{\pi_k}\}$
    for all $\mathbf{y} \in \{\mathbf{y}^{\pi_1}, .., \mathbf{y}^{\pi_k}\}$ **do**
      Compute $\boldsymbol{\mu}_\boldsymbol{\theta}(\mathbf{y}), \boldsymbol{\Sigma}_\boldsymbol{\theta}(\mathbf{y})$.
      Sample $\mathbf{y}_\mathbf{a}$ from $\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_\boldsymbol{\theta}(\mathbf{y}), \boldsymbol{\Sigma}_\boldsymbol{\theta}(\mathbf{y}))$.
    end for
    Compute $\mathrm{L}(\boldsymbol{\theta}) = \mathbb{E}\left[l(\mathbf{y}, \mathbf{y}_\mathbf{a}; \boldsymbol{\phi})\right]$ using the minibatch samples and the corresponding attacks.
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_c \nabla_\boldsymbol{\theta}(\mathrm{L}(\boldsymbol{\theta}))$
    $n \leftarrow n + 1$
  end while
OUTPUT: $\boldsymbol{\theta}$

Random attack mechanism

Another possible way to generate harmful attacks is to create them thanks to a parametric model, called a noisy channel, that will learn the perturbation distribution. If we consider a gaussian noisy channel, we have that $\mathbf{y_a} \sim q_{\boldsymbol{\theta}}(\mathbf{y_a}|\mathbf{y}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{y}), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{y}))$. The problem in Equation 5.15 then becomes:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}:\mathbf{y_a}\sim q_{\boldsymbol{\theta}}(\cdot|\mathbf{y})}{\arg\max}\ \mathbb{E}\left[l(\mathbf{y}, \mathbf{y_a}; \boldsymbol{\phi})\right]. \tag{5.16}$$

The optimization process is explained in Algorithm 3.

The idea behind this method is that we can model the noisy conditional distribution $p_{\mathbf{y_a}|\mathbf{y}}$ as the product of independent normal distributions, where the mean and variance of each component of $\mathbf{y_a}$ depend on the observation $\mathbf{y}$. This can be seen as the generalization of gaussian attacks presented in SUN and collab. [2019] to conditional gaussian attacks. It should be noted that conditional gaussian attacks are much more powerful than gaussian attacks, thus allowing to create more harmful attacks. On the other hand, training and using a parametric model as the noisy channel will allow the attacker to use and share knowledge among all observation samples. We believe sharing this knowledge will enable the attacker to create more potent and less detectable corrupted samples.

5.3.4 Training of robust state estimator

Given that the parameters $\boldsymbol{\phi}$ of a state estimator evolve during training, it is possible to attack the network at each training step. Therefore, to ensure the state estimator is robust, it is convenient to generate attacked samples at each training step and use them to optimize the loss.

Two different steps are needed to train a robust state estimator:

- First, corrupted samples are generated using either the deterministic attack method proposed in Algorithm 2, or the random attack method described in Algorithm 3.
- Then, the defender updates its parameters $\boldsymbol{\phi}$ by approximately solving Equation 5.6.

It should be noted that this process can be repeated until convergence. Using this methodology, we derive an approximate solution to problem in Equation 5.5 leading to our robust estimator. The robust state estimator training is described in Algorithm 4.

Algorithm 4 Training of robust state estimator

INPUT: Define the learning rate α_r and the number of steps n_s
INPUT: input data $\{\mathbf{y}^1, .., \mathbf{y}^m\}$
Initialize the parameters $\boldsymbol{\phi}$
$n = 0$
while $n < n_s$ **do**
Draw a minibatch $\{\mathbf{y}^{\pi_1}, .., \mathbf{y}^{\pi_k}\}$
Generate attacked samples $\mathbf{y_a}$ either according to Algorithm 2 or by performing 1 step of Algorithm 3.
Compute $L(\boldsymbol{\phi})$ from Equation 5.6.
$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \alpha_r . \nabla_{\boldsymbol{\phi}}(L(\boldsymbol{\phi}))$.
$n \leftarrow n + 1$.
end while
OUTPUT: $\boldsymbol{\phi}$

5.4 Experiments

First, we consider a small well-known system - the IEEE 14-bus system - to assess the strength of our proposed attack/defense scheme and to choose all their related hyperparameters. Later, we test our robust state estimator on two additional systems, the IEEE 57 and 118 bus systems.

5.4.1 Set-up

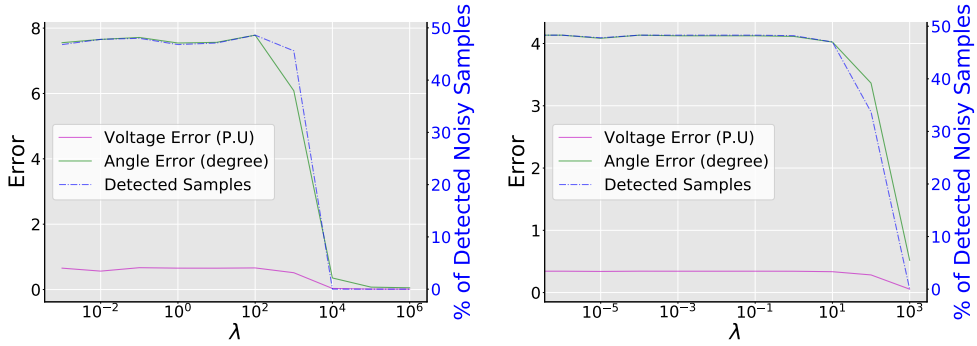
AutoEncoder:

- *Encoder:* We use an encoder composed of 3 dense layers, with ReLU as activation functions.
- *Decoder:* We use a physical knowledge-based decoder, defined using Kirchhoff and Ohm laws.
- *Considered losses:* We need to choose 3 different losses: the attacker's objective $\ell(\cdot,\cdot;\cdot)$, the state estimator reconstruction loss $\mathscr{L}(\cdot,\cdot)$, and the l_p-norm of the bad data detector. We chose the l_1 error as $\mathscr{L}(\cdot,\cdot)$ since it gave us the best l_2-reconstruction error. Therefore, we also use it for the bad data detector. To have the attacker-defender problems as symmetrical as possible, we use the Fisher-Rao distance as the attacker's objective, i.e., $\ell(\hat{\mathbf{x}}, \hat{\mathbf{x}}_\mathbf{a}; \boldsymbol{\phi}) = d_R\big(q_{\boldsymbol{\phi}}(\cdot|\mathbf{y}), q_{\boldsymbol{\phi}}(\cdot|\mathbf{y_a})\big)$.

Optimization set-up: The Stochastic Gradient Descent optimizer is chosen to optimize our VAEs, with a learning rate of 10^{-2}, decayed by 10 after $1/100^{th}$ of the training and by 100 after 1/3 of the training; and a weight decay of 10^{-5}.

Generation of the attacks:

- *Deterministic attack:* We use the method detailed in Algorithm 2 to corrupt samples, using $\delta = 0.1$ and $n = 10$.



(a) Deterministic Attack (b) Random Attack

Figure 5.1: Averaged angle and tension attack-induced errors, and percentage of detected noisy samples under (a) the deterministic attack, (b) the random attack with no defense, as a function of λ, the attack hyperparameter.

- *Random attack:* We use a 5-dense-layer model, with input and output sizes for each layer equal to the number of meters. The chosen optimizer is a Stochastic Gradient Descent optimizer with a learning rate of 10^{-4} and a weight decay of 5.10^{-4}. We perform 1 step of the channel before doing one step of the state estimator during the training phase and reinitialize it at the end of each epoch. We train another channel from scratch for one epoch and discard the detected samples for testing.

5.4.2 Performance of attacks

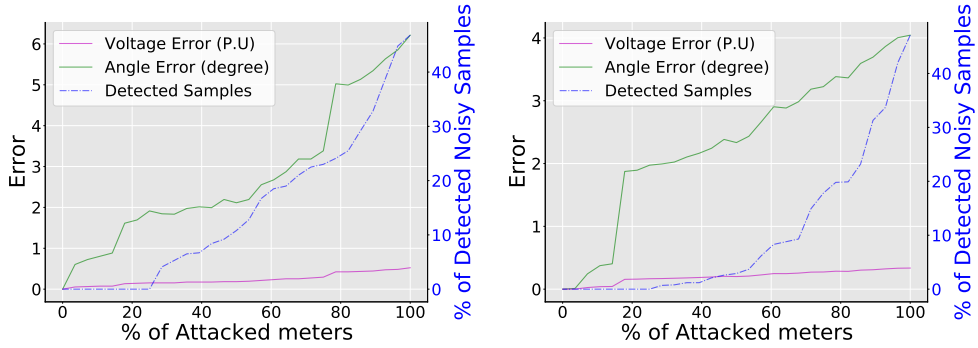
First, we want to assess the strength of the different attack mechanisms. We attack a defenseless state estimator, trained using the set-up presented in Subsection 5.4.1, using both attacks and plot the angle and voltage attack-induced errors along with the percentage of detected samples.

1) Deterministic attack using PGD method

The results for the deterministic attack generation method are presented in Figure 5.1a. For small values of λ, the hyperparameter that controls the trade-off between effectiveness and detection of the attack, the attacks are widely detected, which is uninteresting. For big values of λ, the attacks are undetected but significantly less powerful. A good trade-off between the two phenomena seems to be for $\lambda = 10^3$ since the attack creates a 7 degree and 0.8 P.U. error, while only 45% of the attacks are detected.

2) Random attack using a noisy channel

The results for the random attack generation method are presented in Figure 5.1b. The same behavior as for the deterministic attack is visible in the results. A good trade-off seems to be for $\lambda = 10$, for which the attacks create a 4-degree and 0.4-P.U. error while 47% of the attacks are detected.



(a) Deterministic attack; $\lambda = 10^3$

(b) Random attack; $\lambda = 10$

Figure 5.2: Influence of the percentage of attacked meters on the angle and tension attack-induced errors and percentage of detected samples under (a) the deterministic attack with $\lambda = 1000$, (b) the random attack with $\lambda = 10$ with no defense.

In real-life scenarios, the attacker does not necessarily have access to all the meters. In the following, we will investigate the effect of the number of attacked meters on the attacks.

5.4.3 Influence of the number of attacked meters

In Figure 5.2a and Figure 5.2b, we present the influence of the percentage of attacked meters on our two attack mechanisms, using the λ value chosen previously. The attacker can attack the n% of meters with the highest l_1-norm $\|\mathbf{a}\|\_1$.

In this case, the two attack mechanisms behave similarly. Indeed, the more meters we can attack, the more powerful the attack is, but the more detectable it is. A good trade-off for both the deterministic and the random attack is 25% of attackable meters.

5.4.4 Effect of β on estimation performance

Now we study the influence of the β parameter to increase the robustness of the state estimator to corrupted observations.

1) Deterministic attack using PGD method: We now train a robust state estimator, using the method detailed in Subsection 5.3.4 using the PGD method to generate attacks. The natural and corrupted angle and voltage errors are presented in Figure 5.3a.

As expected, the β hyperparameter controls the trade-off between natural and attacked performances. For small values of β, the state estimator focuses on estimating the natural performances and does not significantly impact the noisy performances. On the opposite, for large values of β, the state estimator focuses on approximating the natural and noisy samples in the same way but no longer tries to estimate the natural observations correctly. Therefore, there exists an optimal β value. In this

particular case, the optimal value of β is equal to 0.03, for which the mean errors are similar for natural and corrupted samples, and their values are 0.27 degrees and 0.015 P.U. on clean samples and 0.016 P.U. on attacked ones, respectively.

2) Random attack using a noisy channel: We perform the same simulations as before, now considering the corrupted samples from the noisy channel's training. The natural and corrupted angle and voltage errors are presented in Figure 5.3b.

The same phenomena as the ones described before are happening in this case. The optimal β value here is also equal to 0.03, which is not surprising since the same defender's objective will be the same no matter how the corrupted samples are generated. When training a defender based on the random attack, we can find a robust VAE for which the mean errors are similar for natural and corrupted samples, and their values are 0.26 degrees and 0.014 P.U. on clean samples, and 0.016 P.U. on attacked ones, respectively.

From all these simulations, we can conclude that using either method to generate attacks, it is possible to create robust state estimators that will have similar state predictions for both natural and attacked samples.

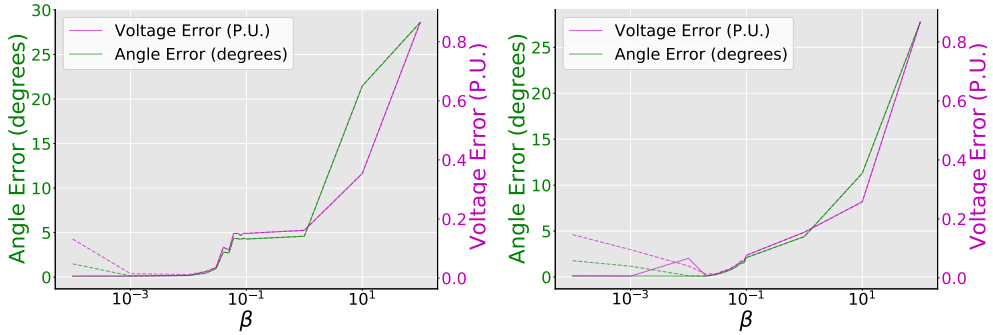
5.4.5 Comparison between the two attack schemes

Finally, to finish comparing the two attack mechanisms, we decided to attack the robust estimator trained with a method using the other. When we attack the robust estimator trained on the PGD method, the robust performances increases to 0.39 degrees as the mean angle error and 0.032 as the mean voltage error. For the other way around, i.e., testing PGD on the estimator trained with the noisy channel, the results are 0.17 degrees and 0.035 P.U., respectively. The two defenses are quite similar. However, training a robust classifier on the noisy channel attack takes approximately three times less time than training a robust classifier on the PGD method. This phenomenon is because it is possible to train the random attacker once per estimator step while we have to complete 10 steps of PGD for every estimator step. In order to increase the power of the deterministic attack, we would have to increase the number of PGD steps, which will be even more computationally costly.

Therefore, we choose the noisy channel attack as the best attack since, while having equivalent performances, it requires a lot less training time.

5.4.6 Comparison between state estimation methods

We now test our method on widely known and used power networks: the IEEE 14/57/118 bus systems. We trained a defenseless, a robust, and a LASSO state estimator following the method presented in Subsection 5.2.3 for each of the datasets and reported the results for natural, random attacked and state-of-the-art (SOTA)



(a) Deterministic attack; $\lambda = 10^3$ (b) Random attack; $\lambda = 10$

Figure 5.3: Natural and attacked angle and tension estimation errors averaged over the buses and the samples for (a) the deterministic attack with $\lambda = 1000$ and 25% of attacked meters, (b) the random attack with $\lambda = 10$ and 25% of attacked meters. Natural values are in plain line, attacked ones are in dashes.

attacked performances presented in Subsection 5.2.2 on Table 5.1. It is clear from this table that our proposed defense mechanism is able to protect our VAE against attack by rightfully estimating the states from both the clean and the noisy observations, whether they are created using our method or the state-of-the-art one. However, the state-of-the-art defense mechanism, using all the hyperparameters defined in the original paper, has more difficulty estimating the clean states and is more sensitive to attacks (i.e., they exhibit a difference of 7 degrees under attacks) than our proposed method.

To conclude, our method seems to be well-suited to craft robust state estimators, for which the state is estimated similarly for clean and attacked samples.

5.5 Conclusion

In this paper, we introduced novel methods to generate powerful false data injection attacks based on the knowledge of the estimator's training procedures. We have experimentally proven that these methods introduce significant state estimation errors. We also introduced a method to learn a robust state estimator using a geometric information distance, known as the Fisher-Rao distance, based on a min-max game between the estimator and the attacker. On multiple benchmarks of power networks, we have experimentally proven that it is possible to train a state estimator that improves its robustness against attacks while not significantly decreasing the performance on clean samples.

Table 5.1: Natural and Attacked Mean Error for Defenseless, Robust and LASSO state estimators for different Bus Systems

Estimator		14buses		57buses		118buses	
		Angle	Voltage	Angle	Voltage	Angle	Voltage
Defenseless	Natural	0.11	0.006	0.12	0.007	0.30	0.003
	Random Attack	2.01	0.17	0.87	0.09	1.34	0.04
	SOTA Attack	2.39	0.21	0.71	0.07	0.41	0.07
Robust	Natural	0.26	0.014	0.21	0.009	0.26	0.003
	Random Attack	0.26	0.016	0.21	0.013	0.26	0.009
	SOTA Attack	0.26	0.014	0.21	0.013	0.28	0.01
LASSO	Natural	30.2	0.050	60.4	0.052	87.4	0.044
	Random Attack	-	-	-	-	-	-
	SOTA Attack	37.4	0.050	69.5	0.054	93.1	0.043

Chapter 5 Conclusion

In this chapter, we addressed the problem of leveraging the knowledge about the output space to increase a state estimator's robustness in the non-linearized case. We presented two procedures to create powerful stealth attack, one deterministic and one random. We compared our two attack frameworks and showed that they were indeed powerful. We used this attacks to craft robust state estimators and experimentally showed that they were efficient against attacks.

5.6 References

ABUR, A. and A. G. EXPOSITO. 2004, *Power system state estimation: theory and implementation*, CRC press. 136, 139

ALDERSON, D. and R. DI PIETRO. 2016, «Operational technology: Are you vulnerable?», *Governance Directions*, vol. 68, nº 6, p. 339–343. 136

ATKINSON, C. and A. F. MITCHELL. 1981, «Rao's distance measure», *Sankhyā: The Indian Journal of Statistics, Series A*, p. 345–365. 142

CELIK, M. K. and A. ABUR. 1992, «A robust wlav state estimator using transformations», *IEEE Transactions on Power Systems*, vol. 7, nº 1, p. 106–113. 137

DÁN, G. and H. SANDBERG. 2010, «Stealth attacks and protection schemes for state estimators in power systems», in *2010 first IEEE international conference on smart grid communications*, IEEE, p. 214–219. 136, 137

GIANNAKIS, G. B., V. KEKATOS, N. GATSIS, S.-J. KIM, H. ZHU and B. F. WOLLENBERG. 2013, «Monitoring and optimization for power grids: A signal processing perspective», *IEEE Signal Processing Magazine*, vol. 30, nº 5, p. 107–128. 139

HENDRICKX, J. M., K. H. JOHANSSON, R. M. JUNGERS, H. SANDBERG and K. C. SOU. 2014, «Efficient computations of a security index for false data attacks in power networks», *IEEE Transactions on Automatic Control*, vol. 59, nº 12, p. 3194–3208. 137

HUG, G. and J. A. GIAMPAPA. 2012, «Vulnerability assessment of ac state estimation with respect to false data injection cyber-attacks», *IEEE Transactions on smart grid*, vol. 3, nº 3, p. 1362–1370. 136, 137

JIN, M., J. LAVAEI and K. JOHANSSON. 2017, «A semidefinite programming relaxation under false data injection attacks against power grid ac state estimation», in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, p. 236–243. 140

JIN, M., J. LAVAEI and K. H. JOHANSSON. 2018, «Power grid ac-based state estimation: Vulnerability analysis against cyber attacks», *IEEE Transactions on Automatic Control*, vol. 64, nº 5, p. 1784–1799. 137, 139

JIN, M., I. MOLYBOG, R. MOHAMMADI-GHAZI and J. LAVAEI. 2019, «Scalable and robust state estimation from abundant but untrusted data», *IEEE Transactions on Smart Grid*, vol. 11, nº 3, p. 1880–1894. 140

KEKATOS, V., G. WANG, H. ZHU and G. B. GIANNAKIS. 2017, «Psse redux: Convex relaxation, decentralized, robust, and dynamic approaches», *arXiv preprint arXiv:1708.03981*. 137

KINGMA, D. P. and M. WELLING. 2014, «Auto-encoding variational bayes», *International Conference on Learning Representations*. 141

KOSUT, O., L. JIA, R. J. THOMAS and L. TONG. 2010, «Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures», in *2010 first IEEE international conference on smart grid communications*, IEEE, p. 220–225. 136, 137

KOTIUGA, W. W. and M. VIDYASAGAR. 1982, «Bad data rejection properties of weughted least absolute value techniques applied to static state estimation», *IEEE Transactions on Power Apparatus and Systems*, , nº 4, p. 844–853. 137

LIANG, G., J. ZHAO, F. LUO, S. R. WELLER and Z. Y. DONG. 2016, «A review of false data injection attacks against modern power systems», *IEEE Transactions on Smart Grid*, vol. 8, nº 4, p. 1630–1638. 137

LIANG, J., O. KOSUT and L. SANKAR. 2014, «Cyber attacks on ac state estimation: Unobservability and physical consequences», in *2014 IEEE PES General Meeting| Conference & Exposition*, IEEE, p. 1–5. 137

LIU, Y., P. NING and M. K. REITER. 2011, «False data injection attacks against state estimation in electric power grids», *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, nº 1, p. 1–33. 136, 137

MADRY, A., A. MAKELOV, L. SCHMIDT, D. TSIPRAS and A. VLADU. 2018, «Towards deep learning models resistant to adversarial attacks», in *International Conference on Learning Representations*. 144

MILI, L., M. G. CHENIAE and P. J. ROUSSEEUW. 1994, «Robust state estimation of electric power systems», *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 41, nº 5, p. 349–358. 137

PASQUALETTI, F., R. CARLI and F. BULLO. 2011, «A distributed method for state estimation and false data detection in power networks», in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, IEEE, p. 469–474. 137

PICOT, M., F. MESSINA, M. BOUDIAF, F. LABEAU, I. B. AYED and P. PIANTANIDA. 2021, «Adversarial robustness via fisher-rao regularization», *arXiv preprint arXiv:2106.06685*. 142

PINELE, J., J. E. STRAPASSON and S. I. COSTA. 2020, «The fisher–rao distance between multivariate normal distributions: Special cases, bounds and applications», *Entropy*, vol. 22, nº 4, p. 404. 142

RAHMAN, M. A. and H. MOHSENIAN-RAD. 2013, «False data injection attacks against nonlinear state estimation in smart power grids», in *2013 IEEE Power & Energy Society General Meeting*, IEEE, p. 1–5. 136

SANDBERG, H., A. TEIXEIRA and K. H. JOHANSSON. 2010, «On security indices for state estimators in power networks», in *First Workshop on Secure Control Systems (SCS), Stockholm, 2010*. 137

SOLTAN, S., M. YANNAKAKIS and G. ZUSSMAN. 2016, «Power grid state estimation following a joint cyber and physical attack», *IEEE Transactions on Control of Network Systems*, vol. 5, nº 1, p. 499–512. 137

SOLTAN, S., M. YANNAKAKIS and G. ZUSSMAN. 2018, «React to cyber attacks on power grids», *IEEE Transactions on Network Science and Engineering*, vol. 6, nº 3, p. 459–473. 137

SOLTAN, S. and G. ZUSSMAN. 2018, «Expose the line failures following a cyber-physical attack on the power grid», *IEEE Transactions on Control of Network Systems*, vol. 6, nº 1, p. 451–461. 137

SOU, K. C., H. SANDBERG and K. H. JOHANSSON. 2013, «On the exact solution to a smart grid cyber-security analysis problem», *IEEE Transactions on Smart Grid*, vol. 4, nº 2, p. 856–865. 137

SUN, K., I. ESNAOLA, S. M. PERLAZA and H. V. POOR. 2019, «Stealth attacks on the smart grid», *IEEE Transactions on Smart Grid*, vol. 11, nº 2, p. 1276–1285. 136, 138, 144, 145

TAJER, A. 2017, «False data injection attacks in electricity markets by limited adversaries: Stochastic robustness», *IEEE Transactions on Smart Grid*, vol. 10, nº 1, p. 128–138. 137

TEIXEIRA, A., K. C. SOU, H. SANDBERG and K. H. JOHANSSON. 2015, «Secure control systems: A quantitative risk management approach», *IEEE Control Systems Magazine*, vol. 35, nº 1, p. 24–45. 137

VOROBEYCHIK, Y. and M. KANTARCIOGLU. 2018, «Adversarial machine learning», *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, nº 3, p. 1–169. 138

WANG, W. and Z. LU. 2013, «Cyber security in the smart grid: Survey and challenges», *Computer networks*, vol. 57, nº 5, p. 1344–1371. 137

XIE, L., Y. MO and B. SINOPOLI. 2010, «False data injection attacks in electricity markets», in *2010 First IEEE International Conference on Smart Grid Communications*, IEEE, p. 226–231. 137

YUAN, Y., Z. LI and K. REN. 2011, «Modeling load redistribution attacks in power systems», *IEEE Transactions on Smart Grid*, vol. 2, nº 2, p. 382–390. 136, 137

ZHANG, Y., R. MADANI and J. LAVAEI. 2017, «Conic relaxations for power system state estimation with line measurements», *IEEE Transactions on Control of Network Systems*, vol. 5, nº 3, p. 1193–1205. 137, 140

ZHU, H. and G. B. GIANNAKIS. 2012, «Robust power system state estimation for the nonlinear ac flow model», in *2012 North American Power Symposium (NAPS)*, IEEE, p. 1–6. 137

Part I Conclusion

Part I focused on the first research question: how can we leverage the information about the internal structure of DNN's output space to improve its robustness? This research question has been addressed for two different applications:

1. **For image data.** We tackled the problem of crafting deep classifiers that are robust to adversarial attackers. We decided to use the Fisher-Rao distance, an information-geometric distance that measures the geodesic between two probability distributions on a statistical manifold. We use this distance as a regularizer to force a given deep classifier to predict natural and adversarial inputs in similar ways. Using the Fisher-Rao distance as a regularizer builds stronger classifiers than using previously introduced regularizers.

2. **For state estimation in Smart Grid systems.** In the critical case of smart grids, where protecting against attacks is crucial to ensure the grid's reliability, we proposed a new framework to build a robust state estimator. We used a similar framework than in Chapter 3 to show that it was possible to construct robust estimators, both under linear and non-linear assumptions. Our contribution is among the first to make use of a deep neural network (*i.e.,* a variational autoencoder in our case) to build defenses against stealth attacks.

Part I was dedicated to increasing our trust in the deep models' decision process by increasing their robustness against attacks.
As previously said, an alternative way to increase our trust in the models is to rely on detecting methods. The detector is an additional module that can be plugged on top of the neural network to protect against malicious agents. Given an input, the detector chooses to reject it based on chosen characteristics. In Part II, we describe our contributions to the field of adversarial detection applied to image data.

Part II

On the use of Simple Statistic Tools to Detect Attacks: Using Data-Depths to Protect the Input's integrity

Part II Abstract

This part is dedicated to our proposed answers to the second question asked in Chapter 1: **How can we craft an efficient and effective detection method based on simple tools ?** This part is split into two chapters consisting of two different contributions.

- In Chapter 6, we present our first solution to craft a detection method. We introduce the data-depths, statistical tools usually used in anomaly detection in the framework of supervised detection of adversarial examples. Data-depths provide a center-outward ordering of points w.r.t. a reference distribution. We specifically use the halfspace-mass depth and design a detection method to leverage the relevant class-wise information present in the different layers of a given model. We also make use of the available knowledge about the possible threats to craft an efficient method to discard attacked inputs. Experimentally, we validate our method in different scenarios: attack-aware, blind-to-attack, and against adaptive attacks, and show the superiority of our method compared to other state-of-the-art detectors.
- In Chapter 7, we present our second and last detection method. In the unsupervised setting, i.e., without any knowledge about the possible threats, we design a detection method that leverages both the internal structure of the Vision Transformers and the available information provided by the use of a specific data-depth: the Integrated Rank-Weighted (IRW) depth. After explaining our different choices, we experimentally prove that our method outperforms state-of-the-art methods and that our method makes the adaptive attacker's job more complicated.

Chapter 6

A Halfspace-Mass Depth-Based Detector for Adversarial Attack Detection

Chapter 6 Abstract

We here present our first contribution concerning the second research question. We made use of the data-depths to detect adversarial examples in a supervised setting. We propose, in the following, a detection method that leverages the particular class-wise information available at different layer of the network , and the halfspace-mass depth to efficiently detect adversarial examples trying to fool an underlying classifier. We experimentally show that the attacks have different class-wise behavior. We compare our method with state-of-the-art detection methods, and experimentally prove its efficiency. Finally, we show that it is more complicated for an adaptive attacker to fool our detection method than others.

Contents

Abstract

Despite the widespread use of deep learning algorithms, vulnerability to adversarial attacks is still an issue limiting their use in critical applications. Detecting these attacks is thus crucial to build reliable algorithms and has received increasing attention in the last few years. In this paper, we introduce the **H**alfsp**A**ce **M**ass de**P**th d**E**tecto**R** (HAMPER), a new method to detect adversarial examples by leveraging the concept of data depths, a statistical notion that provides center-outward ordering of points with respect to (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 directly attack HAMPER via gradient based-methods. We evaluate HAMPER in the context of supervised adversarial attacks detection across four benchmarks datasets. Overall, we empirically show that HAMPER consistently outperforms SOTA methods. In particular, the gains are 13.1% (29.0%) in terms of AUROC↑ (resp. FPR$\downarrow_{95\%}$) on SVHN, 14.6% (25.7%) on CIFAR10 and 22.6% (49.0%) on CIFAR100 compared to the best performing method.

6.1 Introduction

In most machine learning applications, deep models have achieved state-of-the-art performance. However, an important limitation to their widespread use in critical systems is their vulnerability to adversarial attacks [SZEGEDY and collab., 2014], i.e., the introduction of maliciously designed data crafted through minor adversarial

perturbations to deceive a trained model. This phenomenon may lead to disastrous consequences in sensitive applications such as autonomous driving, aviation safety management, or health monitoring systems [GEIFMAN and EL-YANIV, 2019; GEIFMAN and collab., 2019; GUO and collab., 2017; MEINKE and HEIN, 2020].

Over time, a vast literature has been produced on defense methods against adversarial examples [ALDAHDOOH and collab., 2021b; ATHALYE and collab., 2018b; CROCE and HEIN, 2020; ZHENG and collab., 2019]. On the one hand, techniques to train models with improved robustness to upcoming attacks have been proposed in MADRY and collab. [2018]; ZHENG and collab. [2016] or PICOT and collab. [2021]. On the other hand, effective methods to detect adversarial examples given a pre-trained model were reported in KHERCHOUCHE and collab. [2020]; MENG and CHEN [2017] or MA and collab. [2019]. Detection methods for adversarial examples can be mainly grouped into two categories [ALDAHDOOH and collab., 2021b]: *supervised* and *unsupervised* ones. In the *supervised* detection setting, the detector is trained on features extracted from adversarial examples generated according to one or multiple attacks. In particular, the *network invariant model approach* consists of features that are derived from the activation values of the network layers (cf. CARRARA and collab., 2018; LU and collab., 2017 or METZEN and collab., 2017); in the *statistical approach* the features are linked to in-training or out-of-training data distribution/manifold (e.g., maximum mean discrepancy [GROSSE and collab., 2017], PCA [LI and LI, 2017], kernel density estimation [FEINMAN and collab., 2017], local intrinsic dimensionality [MA and collab., 2018], latent graph neighbors [ABUSNAINA and collab., 2021]); in the *auxiliary model approach*, the features are instead

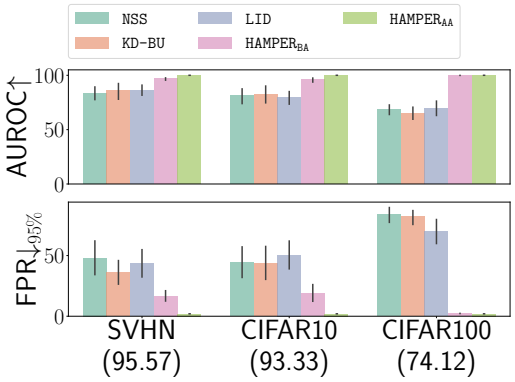


Figure 6.1: Average performances of our method (i.e., `HAMPER`) in an attack-aware and single detector setting, along with the performances of state-of-the-art detection mechanisms (i.e., `NSS`, `LID`, `KD-BU`), on three classically considered datasets (i.e., SVHN, CIFAR10 and CIFAR100). Below the dataset names is the accuracy of their underlying classifiers. In addition to outperforming other methods on all three considered datasets, our method, contrary to the others, does not lose performances as the classifier's accuracy decreases.

derived from monitoring clean and adversarial characteristics (e.g., model uncertainty [FEINMAN and collab., 2017], natural scene statistics [KHERCHOUCHE and collab., 2020]). In the *unsupervised* detection setting, the detector does not rely on the prior knowledge of the attacks, and it only learns from the clean data at training time. Different techniques are used to extract the meaningful features (e.g., *feature squeezing* [LIANG and collab., 2021; XU and collab., 2018], *denoiser approach* [MENG and CHEN, 2017], *network invariant* [MA and collab., 2019], *sensitivity to noise* [HU

and collab., 2019], *auxiliary model* [ALDAHDOOH and collab., 2021a; SOTGIU and collab., 2020; ZHENG and HONG, 2018]). Detection methods of adversarial examples differ as well according to whether the underlying classifier is assumed to be pre-trained or not: when further training of the classifier is allowed, the methods present *a novel training procedure* (e.g., with reverse cross-entropy [PANG and collab., 2018]; with the rejection option [ALDAHDOOH and collab., 2021a; SOTGIU and collab., 2020]) and a thresholding test strategy. Finally, the learning task of the underlying network also impacts the adversarial examples detection methods (e.g., detection of adversarial examples for human recognition tasks [TAO and collab., 2018]).

Adversarial detection can be related to the *anomaly detection* problem. Indeed, anomaly detection aims to identify abnormal observations without previously knowing them, possibly including adversarial attacks. A plethora of techniques has been designed to address this problem ranging from machine learning algorithms such as Isolation Forest [LIU and collab., 2008; STAERMAN and collab., 2019], Local Outlier Factor [BREUNIG and collab., 2000] or One-Class SVM [SCHÖLKOPF and collab., 2001] to statistical tools such as kernel density estimation [FEINMAN and collab., 2017] or Data Depth [ZUO and SERFLING, 2000] (see CHANDOLA and collab. [2009] for an extensive review of anomaly detection methods). In particular, data depth stands out as a natural candidate to detect anomalies [CHEN and collab., 2009].

The idea of statistical depth has grown in popularity in multivariate data analysis since its introduction by John Tukey [TUKEY, 1975]. For a distribution on $\mathbb{R}^d$ with $d > 1$, by transporting the natural order on the real line to $\mathbb{R}^d$, a depth function provides a center-outward ordering of points w.r.t. the distribution. *The higher the point depth score, the deeper the point is in the distribution.* In addition to anomaly detection [CHEN and collab., 2009; SERFLING, 2006; STAERMAN and collab., 2020, 2021b], the notion of depth has been used to extend the notions of (signed) rank or order statistics to multivariate data, which find numerous applications in statistics and machine learning (e.g. robust inference [CUEVAS and collab., 2007], classification [LANGE and collab., 2014], hypothesis testing [OJA, 1983], clustering [JÖRNSTEN, 2004; STAERMAN and collab., 2021a]. To the best of our knowledge, it has not been investigated yet through the lens of adversarial attack detection. This paper aims to leverage this overlooked notion to build an adversarial attack detector.

Contributions. Our contribution is threefold:

1. We propose applying the halfspace-mass depth notion in the context of the adversarial detection problem. To the best of our knowledge, we are the first to both explore and successfully apply data depth for adversarial detection.

2. Through an analysis of the classifier's behavior under threat, we show how to leverage the halfspace-mass depth to build an anomaly score. To that end, we introduce `HAMPER`, a simple supervised method to detect adversarial examples given

a trained model. Given an input sample, HAMPER relies on a linear combination of the halfspace-mass depth score. These depth scores are computed w.r.t. a reference distribution corresponding to the training data conditioned per-class and per-layer.

3. We extensively evaluate HAMPER's performance across popular attack strategies and computer vision benchmark datasets (e.g., SVHN, CIFAR10, and CIFAR100). As shown by Figure 6.1, HAMPER largely outperforms SOTA detection methods and consistently detects attacks that SOTA approaches fail to identify.

The paper is organized as follows. First, in Section 6.2, we describe the adversarial detection problem and provide a detailed overview of the SOTA supervised detection methods and the attack mechanisms considered throughout the paper. In Section 6.3, after recalling the concept of data depth by focusing on the halfspace-mass depth, we introduce HAMPER, our proposed supervised detector method based on the HM depth. In Section 6.4, we provide insights on the underlying classifier's behavior under threats. In Section 6.5, we extensively evaluate HAMPER through numerical experiments on benchmarks on visual datasets and compare it to SOTA methods. Finally, concluding remarks are gathered in Section 6.6.

6.2 Background

After defining the problem formulation, we present the SOTA detection methods and the attack mechanisms that we will consider throughout this paper.

6.2.1 Problem formulation

Let $(\mathbf{X}, \mathrm{Y})$ be a random tuple of variables valued in $\mathscr{X} \times \mathscr{Y}$ with unknown data distribution $p_{\mathbf{X}\mathrm{Y}}$; $\mathscr{X} \subset \mathbb{R}^d$ represents the feature space and $\mathscr{Y} = \{1, \ldots, \mathrm{C}\}$ represents the labels attached to elements in $\mathscr{X}$, where $\mathrm{C} \in \mathbb{N}$, $\mathrm{C} \geq 2$. The training dataset $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is defined as $n \geq 1$ independent identically distributed (i.i.d.) realizations of $p_{\mathbf{X}\mathrm{Y}}$. Subsets of the feature space associated with a label $c \in \mathscr{Y}$ are denoted by $\mathscr{S}_c = \{\mathbf{x}_i \in \mathscr{S} : y_i = c\}$ with $\mathscr{S} = \{\mathbf{x}_i\}_{i=1}^n$.

Given a parametric model with $\mathrm{L} \geq 1$ layers; let $f_{\boldsymbol{\theta}}^{\ell} : \mathscr{X} \to \mathbb{R}^{d_\ell}$ with $\ell \in \{1, \ldots, \mathrm{L}\}$, denotes the output of the ℓ-th layer of the deep neural network (DNN) parametrized by $\boldsymbol{\theta} \in \Theta$ where the dimension of the latent space induced by the ℓ-th layer is d_ℓ. The class prediction is obtained from the L-th layer softmax output as follows:

$$f_{\boldsymbol{\theta}}^{\mathrm{L}}(\mathbf{x}; \mathscr{D}) \triangleq \arg\max_{c \in \mathscr{Y}} q_{\boldsymbol{\theta}}(c|\mathbf{x}) \text{ with } q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) = \mathrm{softmax}(f_{\boldsymbol{\theta}}^{\mathrm{L}-1}(\mathbf{x}; \mathscr{D})).$$

The adversarial problem. Given $\mathbf{x} \in \mathscr{X}$ and $p \geq 1$, the adversarial generation problem

can be defined as producing $\mathbf{x}'$ such as [SZEGEDY and collab., 2014]:

$$\mathbf{x}' = \underset{\mathbf{x}' \in \mathbb{R}^d : \|\mathbf{x}' - \mathbf{x}\|_p < \varepsilon}{\arg\min} \|\mathbf{x}' - \mathbf{x}\| \text{ s.t. } f^{\mathrm{L}}_{\boldsymbol{\theta}}(\mathbf{x}'; \mathscr{D}) \neq y, \tag{6.1}$$

where y is the true label associated to the sample $\mathbf{x}$, and $\|\cdot\|_p$ is the p-norm operator. Since this problem is computationally infeasible in general, it is commonly relaxed as follows:

$$\mathbf{x}' = \underset{\mathbf{x}' \in \mathbb{R}^d : \|\mathbf{x}' - \mathbf{x}\|_p < \varepsilon}{\arg\max} \mathscr{L}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}), \tag{6.2}$$

where $\mathscr{L}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$ is the objective of the attacker, representing a surrogate of the constraint to fool the classifier, i.e., $f^{\mathrm{L}}_{\boldsymbol{\theta}}(\mathbf{x}'; \mathscr{D}) \neq y$. The variety of attacks differs with the choice of the norm (e.g., $p = 1, 2, \infty$) and the value of ε.

Crafting a detector. Given a new observation $\mathbf{x} \in \mathbb{R}^d$, detecting adversarial attacks boils down to build a binary rule $g : \mathbb{R}^d \to \{0, 1\}$. Namely: a new observation $\mathbf{x} \in \mathbb{R}^d$ is considered as 'normal' (or 'natural', 'clean'), i.e. generated by $p_{\mathbf{X}\mathrm{Y}}$, when $g(\mathbf{x}) = a$ with $a \in \{0, 1\}$, and $\mathbf{x}$ is considered as an adversarial example when $g(\mathbf{x}) = 1 - a$. For a given scoring function $s : \mathbb{R}^d \to \mathbb{R}$, and a threshold $\gamma \in \mathbb{R}$, we have

$$g(\mathbf{x}) = \mathbb{I}\{s(\mathbf{x}) > \gamma\} = \begin{cases} 1 \text{ if } s(\mathbf{x}) > \gamma, \\ 0 \text{ if } s(\mathbf{x}) \leq \gamma. \end{cases} \tag{6.3}$$

6.2.2 Supervised detection methods

Supervised detection methods, when the defender has access to the future threats that it is going to face, can be separated into two main groups: attack-aware and blind-to-attack methods.

Attack-aware methods. In the attack-aware setting, the methods are going to face a single threat, and they have full knowledge about them. Therefore, it is possible to train one detector per attack. In this setting fall two detection methods: `KD-BU` [FEINMAN and collab., 2017] and `LID` [MA and collab., 2018]. `KD-BU` is based on the intuition that the adversarial examples lie off the data manifold. To train the detector, a *kernel density* estimation in the feature space of the last hidden layer is performed, followed by an estimation of the *bayesian uncertainty* of the input sample. `LID` extracts the *local intrinsic dimensionality* features for natural and attacked samples for each layer of the classifier and trains a detector on them.

Blind-to-Attack setting. In the blind-to-attack setting, the defender has knowledge about the fact that it is going to be attacked, but do not know exactly how. In that case, a single detector is trained, deployed and tested against all possible threats. `NSS` [KHERCHOUCHE and collab., 2020] falls into that category. It is based on the

extraction of the *natural scene statistics* from the clean and adversarial samples from different threats models, later used to train a detector to discriminate between natural inputs and adversarial examples. Natural scene statistics are regular statistical properties that are altered by adversarial perturbations.

6.2.3 A brief review of attack mechanisms

Multiple methods to generate adversarial examples have been developed in recent years. The attack mechanisms can be divided into two main categories: whitebox, where the attacker has complete knowledge about the targeted classifier, and blackbox, where the attacker does not know about the targeted classifier.

Whitebox attacks. The simplest one is Fast Gradient Sign Method (**FGSM**), introduced by Goodfellow *et al.* [GOODFELLOW and collab., 2015]. It consists in modifying the examples in the direction of the gradient of a specific objective, w.r.t. the input on the targeted classifier. Two iterative versions of FGSM have been proposed: Basic Iterative Method (**BIM**; KURAKIN and collab., 2018) and Projected Gradient Descent (**PGD**; MADRY and collab., 2018). The main difference is that BIM initializes the adversarial example to the natural sample while PGD initializes it to the natural example plus random noise. Although PGD was initially created under an L_∞ constraint, it is possible to extend the method to any L_p-norm constraint. Later, Moosavi-Dezfooli *et al.* [MOOSAVI-DEZFOOLI and collab., 2016] introduced DeepFool (**DF**), an iterative method based, at each step, on a local linearization of the model, resulting in a simplified problem. Finally, Carlini-Wagner [CARLINI and WAGNER, 2017] presents the **CW** method to find the smallest noise solving the original adversarial problem. They proposed a new relaxed version of the adversarial problem that optimizes an attack objective, chosen according to a specific task.

Blackbox attacks. Without any knowledge about the targeted classifier or its gradients, blackbox attacks are expected to rely on different mechanisms. Square Attack (**SA**; ANDRIUSHCHENKO and collab., 2020) employs a random search for perturbations that maximize a given objective, Spatial Transformation Attack (**STA**; ENGSTROM and collab., 2019) applies small translations and rotations to the original image while Hop Skip Jump (**HOP**; CHEN and collab., 2020) estimates the gradient-based direction to perturb through a query on the targeted classifier.

Adaptive attacks. There exists a third type of attacks called **Adaptive Attacks** [ATHALYE and collab., 2018a; CARLINI and WAGNER, 2017; TRAMER and collab., 2020; YAO and collab., 2021]. Adaptive attacks have full knowledge about not only the underlying classifier to attack, but also the defense mechanisms one may have deployed.

To build efficient adaptive attacks, it is therefore crucial to understand the mechanisms involved into the defense, and finding ways to bypass them. For examples, the Backward-Pass Differentiable Attack (**BPDA**; ATHALYE and collab., 2018a) has been developed to overcome the non-differentiability of the defense mechanisms by finding a suitable surrogate to the non-differential parts of them.

6.3 A Depth-Based Detector

After presenting the data depth in Subsection 6.3.1, with an emphasis on the halfspace-mass depth, we introduce our depth-based detector in Subsection 6.3.2.

6.3.1 Background on data-depth

A data depth function $D(\cdot, P) : \mathbb{R}^d \to [0,1]$ measures the centrality of any element in $\mathbf{x} \in \mathbb{R}^d$ w.r.t. a probability distribution P (respectively, a data set). It provides a center-outward ordering of points in the support of P and can be straightforwardly used to extend the notions of rank or order statistics to multivariate data. The higher $D(\mathbf{x}, P)$, the deeper $\mathbf{x} \in \mathbb{R}^d$ is in P. The earliest proposal is the *halfspace* depth introduced by John Tukey in 1975 [TUKEY, 1975]. This depth is very popular due to its appealing properties and ease of interpretation. Assume that P is defined on an arbitrary subset $\mathscr{K} \subset \mathbb{R}^d$ and denote by $P(H) \triangleq P(H \cap \mathscr{K})$ the probability mass of the closed halfspace H. The halfspace depth of a point $\mathbf{x} \in \mathbb{R}^d$ with respect to a probability distribution P on $\mathbb{R}^d$ is defined as the smallest probability mass that can be contained in every closed halfspaces containing $\mathbf{x}$:

$$D_H(\mathbf{x}, P) = \inf_{H \in \mathscr{H}(\mathbf{x})} P(H), \tag{6.4}$$

where $\mathscr{H}(\mathbf{x})$ is the set of all closed halfspaces containing $\mathbf{x}$.

However, the halfspace depth suffers from three critical issues: (i) finding the direction achieving the minimum to assign it a score induces a significant sensitivity to noisy directions, (ii) assigning the zero score to each new data point located on the outside of the convex hull of the support of P makes the score of these points indistinguishable, and (iii) as the dimension of data increases, an increasing percentage of points will appear at the edge of the convex hull covering the data set leading to have low scores to every points.

To remedy those drawbacks, alternative depth functions have been independently introduced in CHEN and collab. [2015] and RAMSAY and collab. [2019]. In this regard, the extension of Tukey's halfspace depth, recently introduced and referred to as the halfspace-mass (HM) depth [CHEN and collab., 2015] (see also RAMSAY and collab., 2019 and STAERMAN and collab., 2021b), offers many advantages.

Authors proposed to replace the infimum by an expectation over all possible closed halfspaces containing $\mathbf{x}$, following in the footsteps of CUEVAS and FRAIMAN [2009]. More precisely, given a random variable $\mathbf{X}$ following a distribution P and a probability measure Q on $\mathscr{H}(\mathbf{x})$, it is defined as follows:

$$\mathrm{D}_{\mathrm{HM}}(\mathbf{x}, \mathrm{P}) = \mathbb{E}_{\mathrm{H}\sim\mathrm{Q}}\left[\mathrm{P}(\mathrm{H})\right]. \tag{6.5}$$

In addition to basic properties a depth function should satisfy, the halfspace-mass depth possesses robustness properties: it has a unique (depth-induced) median with an optimal breakdown point equal to 0.5 [CHEN and collab., 2015] which means that the halfspace-mass depth provides a stable ordering of the 'normal' data even when polluted data belong to the training set. In addition, it has been successfully applied to anomaly detection in CHEN and collab. [2015] making it a natural choice to adversarial attack detection. When a training set $\{\mathbf{x}_i, \ldots, \mathbf{x}_n\}$ is given, Equation 6.5 boils down to:

$$\mathrm{D}_{\mathrm{HM}}(\mathbf{x}, \mathrm{P}_n) = \mathbb{E}_{\mathrm{Q}}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\{\mathbf{x}_i \in \mathrm{H}\}\right], \tag{6.6}$$

where P_n is the empirical measure defined by $\frac{1}{n}\sum_{i=1}^{n}\delta_{\mathbf{x}_i}$. The expectation can be conveniently approximated with a Monte-Carlo scheme in contrast to several depth functions that are defined as solutions of optimization problems, possibly unfeasible in high dimension. The aim is then to approximate Equation 6.6 by drawing a finite number of closed halfspaces containing $\mathbf{x}$ (see Section B.1 for the approximation algorithm for training and testing).

6.3.2 Our Depth-Based Detector

The methodology we propose here is based on the halfspace-mass depth that exhibits attractive mathematical and computational properties, as described in the previous section.

Our depth-based detector, `HAMPER`, relies on the information available in a subset Λ of DNNs' layers, i.e., the mapped data $\mathbf{z}_{\ell,i} = f_{\boldsymbol{\theta}}^{\ell}(\mathbf{x}_i; \mathscr{D})$, $\ell \in \Lambda \subset \{1, \ldots, \mathrm{L}-1\}$. We denote by $\widetilde{\mathscr{P}}^{\ell} = \{\mathbf{z}_{\ell,i}\}_{i=1}^{n}$ and $\widetilde{\mathscr{P}}_c^{\ell} = \left\{\mathbf{z}_{\ell,i} \in \widetilde{\mathscr{P}}^{\ell} : y_i = c\right\}$, the ℓ-th and the ℓ-th class-conditionally representations of the training dataset, respectively. Our approach aims to construct a score function $s : \mathbb{R}^d \to [0,1]$ providing a confidence level to a new observation $\mathbf{x}$ indicating its degree of abnormality w.r.t. to the training dataset. `HAMPER` leverages appealing properties of the HM depth detailed in Subsection 6.3.1 and can be summarized into two distinct steps. The function s is built by first constructing $|\Lambda| \times \mathrm{C}$ intermediate scoring functions $s_{\ell,c} : \mathbb{R}^{d_\ell} \to [0,1]$ designed for each considered layer and each class. The map $s_{\ell,c}$ assigns a value to any element of the

embedded space of the ℓ-th layer represented somehow its 'distance' to the class c of the mapped training set. Thereafter, an aggregation is performed between scores using a small validation dataset composed of both 'normal' and 'adversarial' samples. These two parts of the proposed approach are detailed below.

Intermediate score functions. Given a new observed image $\mathbf{x} \in \mathbb{R}^d$ mapped into $|\Lambda|$ representations $\{\mathbf{z}_\ell\}_{\ell \in \Lambda}$ such that $\mathbf{z}_\ell = f_{\boldsymbol{\theta}}^{\ell}(\mathbf{x}; \mathscr{D})$, we propose to use the HM depth as intermediate scoring functions $s_{\ell,c}$. Precisely, we compute $\mathrm{D}_{\mathrm{HM}}(\mathbf{z}_\ell, \widetilde{\mathscr{S}}_c^{\ell})$, for each considered layer ℓ and each class c, i.e., the HM depth between $\mathbf{z}_\ell = f_{\boldsymbol{\theta}}^{\ell}(\mathbf{x}; \mathscr{D})$ and the class-conditionally probability distribution of the training dataset $\widetilde{\mathscr{S}}_c^{\ell} = \{f_{\boldsymbol{\theta}}^{\ell}(\mathbf{x}_i; \mathscr{D}) : \mathbf{x}_i \in \mathscr{S}_c\}$. Following the approximation algorithm of the HM introduced in CHEN and collab. [2015], we use an efficient training/testing procedure in order to compute D_{HM} (summarized in Algorithm 5 and Algorithm 6 in Section B.1). These algorithms are repeated for each class c and each considered layer ℓ leading to $|\Lambda| \times \mathrm{C}$ scoring functions. Three parameters with low sensitivity are involved: K which is the number of sampled halfspaces in order to approximate the expectation of Equation 6.5; the size n_s of the sub-sample drawn at each projection step; and the λ hyperparameter which controls the extent of the choice of the hyperplane. In this paper, we follow the advice given in CHEN and collab. [2015] by choosing the following parameters $\mathrm{K} = 10000$, $n_s = 32$ and $\lambda = 0.5$ offering a good compromise between performance and computational efficiency.

Aggregation procedure. Following the supervised setting scenario, as in FEINMAN and collab. [2017]; KHERCHOUCHE and collab. [2020] or MA and collab. [2018], the score is obtained through an aggregation which is performed between halfspace-mass scores using a small validation dataset composed of 'normal' and 'adversarial' samples. Our scoring function is then formally defined as:

$$s(\mathbf{x}) = \sum_{\ell \in \Lambda} \sum_{c=1}^{\mathrm{C}} \alpha_{\ell,c} \, \mathrm{D}_{\mathrm{HM}}(\mathbf{z}_\ell, \widetilde{\mathscr{S}}_c^{\ell}), \tag{6.7}$$

where the weights $\alpha_{\ell,c}$ are obtained through the training of a linear regressor in a supervised manner. It is worth noting that the anomaly score s from Equation 6.7 results from both class and layer dependent linear combination. The class dependency of s is motivated by (1) the per class behavior classifier which is highlighted in Subsection 6.4.2 and by (2) the monotonicity relative to deepest point property of the halfspace-mass depth and depth functions in general (see e.g. property ($\mathbf{D}_3$) in Section B.2 of the Appendix or STAERMAN [2022]; ZUO and SERFLING [2000]). The layer dependency of s is motivated by the per layer behavior classifier which is displayed in Subsection 6.4.3.

Referring to the problem formulation notations (see Subsection 6.2.1), given a

threshold γ, and supposing $a = 1$, the detector is provided by the Equation 6.3. The overview of HAMPER can be summarized in Algorithm 7 in Section B.1.

6.3.3 Comparison to state-of-the-art detection methods.

We benchmark our approach with three supervised detection methods: LID, KD-BU, and NSS. We chose these baselines because they are supervised and do not modify the model to protect. We could consider ABUSNAINA and collab. [2021] but the codes are not publically available.

Local Intrinsic Dimensionality (LID). LID is based on the intuition that adversarial examples lie outside of the clean data manifold. By computing the Local Intrinsic Dimensionality, it is possible to check whether the new point is close to the original data manifold. Following this idea, three version of each natural samples are used. The clean one, a noisy version of it, and an attacked one. The LID approximate for each of those points to the clean distribution will be computed at the output of each layers. Those variables will then be used to train a detector that will distinguish between adversarial samples, and normal ones (normal samples are either clean or noisy samples). Each of the strategy to craft adversarial samples can have very different LID characteristics, a detector per type of threats is therefore necessary. LID therefore lies in the *attack-aware* category.

Kernel Density and Bayesian Uncertainty (KD-BU). KD-BU also relies on the idea that adversarial samples lie off the original data manifold. To detect adversarial samples, they first perform a kernel density estimate at the last hidden layer level to detect samples that are far from the original manifold. Then, a bayesian uncertainty estimate is computed to detect when points lie in low-confidence regions of the input space. Both of those characteristics are later used to train a detector that distinguish between natural and adversarial examples. Once again, the kernel density estimates and the bayesian uncertainty values for different types of attacks can differ a lot, therefore this method have been created to be *attack-aware*.

Natural Scene Statistics (NSS). NSS relies on the extraction of the natural scene statistics at the image level. Natural scene statistics are statistics that will be very different for natural and attacked images. Indeed, for clean image, applying the natural scene statistics will output an image with meaning, however, for attacked samples, the resulting image will have no meaning. The Natural Scene Statistics extraction is then used to train a detector to distinguish between natural and attacked samples. To overcome the need to have a specific detector per attack, the authors of NSS decided to train their detector using the natural scene statistics of various attacks. It therefore lies in the *blind-to-attack* category.

HAMPER. Our proposed HAMPER detector is computing, thanks to the halfspace-mass depth, the distance of a given $\mathbf{x}$ to a reference training distribution. In the

sense that it compares between a novel point and a reference, our method is close to the LID method, however, as explained in the previous section, we do not compute our anomality score (i.e., the HM depth) at each layer's output, but we only use a subset of layers Λ. In addition, it is possible to use our proposed detector under both scenarios, i.e., the *attack-aware* and the *blind-to-attack* scenarios.

6.4 Analyzing Statistical Information of the Networks' Behavior under Threats

In this section, we provide insights into the attackers' and defenses' behavior from the classifier's perspective. This section aims to provide justification and insights on the choice of making the linear weights of Equation 6.7 dependant on both the class and the layer. In particular, we analyze the behavior of the classifier on attacks in Subsection 6.4.2, while in Subsection 6.4.3 we explore which subset of layers of the classifier carries the relevant information to build an efficient supervised data-depth based detector.

6.4.1 Experimental setting

Datasets and classifiers. We run our experiments on three image datasets: SVHN [NETZER and collab., 2011], CIFAR10 and CIFAR100 [KRIZHEVSKY, 2009]. We train a classifier that aims at rightfully classifying natural examples for each of those datasets. For SVHN and CIFAR10 we use a ResNet-18 trained for 100 epochs, using an SGD optimizer with a learning rate of 0.1, weight decay of 10^{-5}, and a momentum of 0.9; for CIFAR100 we chose a ResNet-110 pre-trained using an SGD optimizer with a learning rate of 0.1, weight decay of 10^{-5}, and a momentum of 0.9. Once trained, all classifiers are frozen.

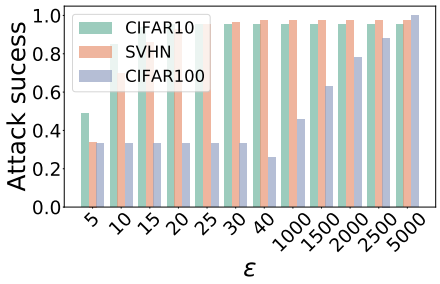
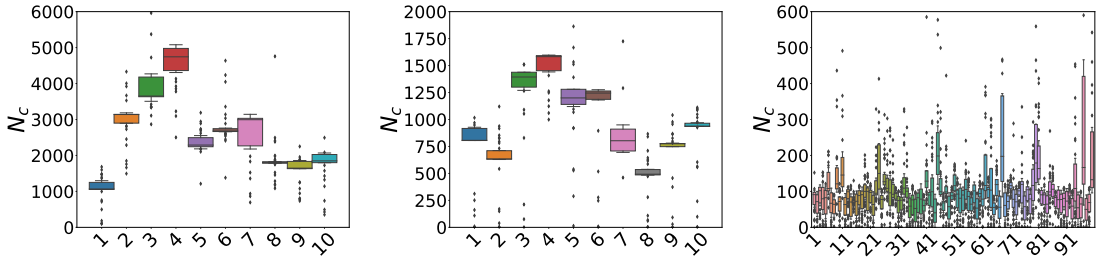


Figure 6.3: **Calibrating the maximal allowed perturbation ε on CIFAR100**. Accuracy on adversarial examples created using PGD_1 for the SVHN, CIFAR10 and CIFAR100 classifiers. *On CIFAR100, to ensure high successes of the attacks, one must allow the attacker to have larger values of ε, compared to the CIFAR10 and SVHN ones.*

Attacks & choice of the maximal allowed perturbation ε. To have a wide range of attacks to test, we use all the methods mentioned in Subsection 6.2.3. For FGSM, BIM and PGD, we consider the L_∞-norm, with multiple ε in {0.0315, 0.0625, 0.125, 0.25, 0.3125, 0.5}. We also generate perturbed examples using PGD under the L_1-

https://github.com/bearpaw/pytorch-classification



(a) SVHN (b) CIFAR10 (c) CIFAR100

Figure 6.2: **Per class behavior analysis.** Average number of adversarial examples per class on each of the considered datasets.

norm constraint with ε varying in {5, 10, 15, 20, 25, 30, 40}, for CIFAR10 and SVHN, and in {40, 500, 1000, 1500, 2000, 2500, 5000} for CIFAR100. Moreover, we generate perturbed examples using PGD under the L_2-norm constraint with ε varying in {0.125, 0.25, 0.3125, 0.5, 1, 1.5, 2}, for CIFAR10 and SVHN, and in {5, 10, 15, 20, 30, 40, 50} for CIFAR100. In order to attack with PGD (L_1 and L_2 norm) the classifier trained on CIFAR100, we chose different epsilon values than those used for CIFAR10 and SVHN since the attacks generated with those epsilons were not able to fool the network (see Figure 6.3). CW attacks are generated under the L_∞ and L_2 constraint, with ε equals to 0.3125 and 0.01 respectively. Finally, we perturb samples using DF which is an L_2 attack without any constraint on ε. Concerning blackbox attack, SA is an L_∞-norm attack $\varepsilon = 0.125$, HOP is an L_2 attack with $\varepsilon = 0.1$. Finally, STA is not concerned by a norm constraint nor a maximal perturbation, the attacker strength is limited in rotation (maximum of 30^o) and in translation (maximum of 8 pixels).

6.4.2 Analyzing the networks' per class behavior under threats

In this section, we investigate the per-class behavior of the image classifier to motivate and justify the choice of *the class dependency of the proposed aggregation procedure* (see Equation 6.7).

Simulation. We examine the distribution of the adversarial examples w.r.t. the class predicted by the classifier. For this purpose, in Figure 6.2, we plot the distribution of adversarial samples per predicted class (N_c) as a function of the class (c).

Analysis. In SVHN, CIFAR10 and CIFAR100 natural images are balanced. However, on these datasets, the per-class distribution of the adversary is not uniform over the classes: in both SVHN (Figure 6.2a) and CIFAR10 (Figure 6.2b), classes 3 and 4 are overly represented on average, on the contrary of class 1 and 8 for SVHN and CIFAR10 respectively. Similarly, in CIFAR100 (Figure 6.2c), the classes 11, 24, 45, 68, 80, 81, and 97 are the overrepresented whilst the classes 37, 59, 65, 67, 76 and 98 are the most underrepresented on average. Note that the diamond points in the plots denote the

outliers, i.e., adversarial examples behaving differently from the others.

Takeaways. The variability of the per-class behavior of the classifier under threats suggests that class is an important characteristic and should be leveraged to detect adversaries. This observation further motivates the per-class computation of the halfspace-mass depths and then the class dependency of the linear regressor of Equation 6.7.

6.4.3 Analyzing the networks' per layer behavior under threats

In this section, we investigate the per-layer behavior of the image classifier to motivate and justify the choice of *the layer behavior of the proposed aggregation procedure* (see Equation 6.7).

Simulation. We investigate each layer's roll on HAMPER 's decision process to better understand each layer importance. In particular, we focus on CIFAR10 with ResNet-18 and we train a linear least squares regressor with L_2 regularization (see e.g. HASTIE and collab., 2009) on the depth features extracted from all the layers, $\Lambda \in \{1,\dots,L-1\}$, of the classifier. In Figure 6.4, we report the weights associated with each layer $\ell \in \Lambda$ of the underlying classifier, averaged over the classes, when changing the values of the L_2 weight constraint (the values are reported in the legend).

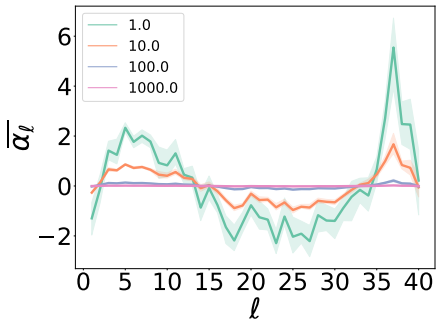


Figure 6.4: **Per layer behavior analysis.** Evolution of $\bar{\alpha}_\ell = \frac{1}{C}\sum_c \alpha_{\ell,c}$ the average and the standard deviation over the classes of the regressor weight as a function of the layer, for different value of the regularization parameter.

Analysis. From Figure 6.4, we observe of the decision of the detector is based on several layers. This a-posteriori analysis justifies the layer dependency of the linear regressor weights. As the L_2 weight regularization increases, the last layers receive more weights, suggesting they are good candidates to build a detector. Although it is possible to rely on the whole set of layers, motivated by efficiency and Figure 6.4, we select the 11 final layers of the classifier, i.e., we base our experiments on the subset $\Lambda = \{L-12,\dots,L-1\}$.

Takeaways. Through Figure 6.4, we have motivated both the per-layer computation of the halfspace-mass depths and then the per-layer dependency of the linear regressor of Equation 6.7.

6.5 Experiments

In this section, we assess the effectiveness of our proposed depth-based detection method. The code will be made available at `github.com`. This section is organized as follows: we first describe the experimental setting in Subsection 6.5.1 and then we provide a detailed discussion of the results in Subsection 6.5.2.

6.5.1 Experimental setting

We refer to Subsection 6.5.1 for the datasets, the classifiers and the attacks we considered for our evaluation.

Evaluation metrics. For each threat scenario, the performance is measured in terms of two metrics:

AUROC↑ (higher is better): the *Area Under the Receiver Operating Characteristic curve* (ROC; DAVIS and GOADRICH, 2006) represents the relation between *True Positive Rate* (TPR) - i.e. adversarial examples detected as adversarial - and *False Positive Rate* (FPR) - i.e. natural samples detected as adversarial. As can be checked from elementary computations the AUROC↑ corresponds to the probability that an natural example has higher score than an adversary sample.

FPR at 95% TPR↓ or FPR$\downarrow_{95\%}$ (lower is better): represents the percentage of natural examples detected as adversarial when 95% of the adversarial examples are detected. The FPR$\downarrow_{95\%}$ is of high interest in practical applications.

Remark. The ideal classifier would reach 100% of AUROC↑ and 0% of FPR$\downarrow_{95\%}$.

6.5.2 Detecting adversarial examples

We recall from Subsection 6.2.2 that we distinguish between two settings in the supervised context: the *attack-aware scenario* and the *blind-to-attack scenario*. Therefore, we conduct two sets of experiments. In the attack-aware scenario, for each attack we train a detector on a validation set - composed of the first 1000 samples of the testing set - and tested on the remaining samples. We refer here to HAMPER-Attack-Aware (HAMPER$_{\text{AA}}$) and we compare it with LID and KD−BU. In the blind-to-attack scenario, we train a unique detector and we test it on all the possible attacks. We refer here to HAMPER-Blind-to-Attack (HAMPER$_{\text{BA}}$) and we compare it with NSS. Note that, while our competitors assign to each input sample the probability of being adversarial, i.e., adversarial samples are labeled as 1, in HAMPER the detector outputs the depth score. This means that a high score corresponds to a deep sample, i.e., a natural one, hence clean samples are labeled as 1.

Table 6.1: Attack-aware performances on the three considered datasets - SVHN, CIFAR10 and CIFAR100 - of HAMPER$_{AA}$ detector together with the results of the SOTA detection methods: LID, and KD−BU, averaged over the L_p-norm constraint. The best results among the detectors are shown in **bold**. The results are presented as AUROC↑ $_{\pm \text{FPR}\downarrow_{95\%}}$ % and in terms of mean (μ) and standard deviation (σ).

		LID			KD−BU			HAMPER$_{AA}$		
		SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100
Norm L_1	μ	64.9 $_{\pm 90.1}$	57.9 $_{\pm 87.7}$	77.6 $_{\pm 47.4}$	84.7 $_{\pm 58.7}$	71.0 $_{\pm 68.3}$	69.5 $_{\pm 71.2}$	**100** $_{\pm 0.0}$	**100** $_{\pm 0.0}$	**100** $_{\pm 0.0}$
	σ	9.2 $_{\pm 5.7}$	11.3 $_{\pm 8.6}$	23.4 $_{\pm 35.7}$	7.0 $_{\pm 16.5}$	24.1 $_{\pm 33.4}$	21.1 $_{\pm 31.1}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$
Norm L_2	μ	78.3 $_{\pm 73.7}$	66.7 $_{\pm 77.3}$	66.5 $_{\pm 65.5}$	84.3 $_{\pm 42.3}$	70.5 $_{\pm 62.2}$	56.2 $_{\pm 87.6}$	**100** $_{\pm 0.0}$	**100** $_{\pm 0.0}$	**100** $_{\pm 0.0}$
	σ	10.0 $_{\pm 16.9}$	15.0 $_{\pm 22.3}$	24.7 $_{\pm 32.8}$	16.2 $_{\pm 31.3}$	26.9 $_{\pm 40.9}$	17.0 $_{\pm 10.3}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$
Norm L_∞	μ	97.5 $_{\pm 9.9}$	94.2 $_{\pm 21.2}$	66.9 $_{\pm 79.1}$	87.3 $_{\pm 21.9}$	93.1 $_{\pm 20.3}$	67.1 $_{\pm 80.3}$	**100** $_{\pm 0.0}$	**100** $_{\pm 0.0}$	**100** $_{\pm 0.0}$
	σ	3.7 $_{\pm 14.0}$	7.6 $_{\pm 24.0}$	17.3 $_{\pm 17.3}$	27.4 $_{\pm 29.9}$	16.8 $_{\pm 32.6}$	15.4 $_{\pm 11.3}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$
No Norm	μ	99.1 $_{\pm 4.4}$	91.7 $_{\pm 36.6}$	98.4 $_{\pm 4.2}$	92.8 $_{\pm 21.9}$	81.4 $_{\pm 76.2}$	76.1 $_{\pm 61.3}$	**100** $_{\pm 0.0}$	**100** $_{\pm 0.0}$	**100** $_{\pm 0.0}$
	σ	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$
Average	μ	86.5 $_{\pm 41.3}$	79.7 $_{\pm 48.6}$	77.4 $_{\pm 49.0}$	86.2 $_{\pm 34.1}$	82.8 $_{\pm 41.6}$	64.9 $_{\pm 80.1}$	**100** $_{\pm 0.0}$	**100** $_{\pm 0.0}$	**100** $_{\pm 0.0}$
	σ	14.8 $_{\pm 38.2}$	18.7 $_{\pm 36.6}$	21.4 $_{\pm 30.4}$	21.8 $_{\pm 31.5}$	23.8 $_{\pm 41.5}$	17.7 $_{\pm 17.7}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$	**0.0** $_{\pm 0.0}$
Global	μ	81.2 $_{\pm 46.3}$			78.0 $_{\pm 51.9}$			**100** $_{\pm 0.0}$		
	σ	19.7 $_{\pm 37.0}$			23.2 $_{\pm 37.6}$			**0.0** $_{\pm 0.0}$		

Attack-aware scenario

Here, we study the performance of the different detectors in the attack-aware scenario and show that HAMPER$_{AA}$ outperforms existing detectors.

Global analysis. We present the attack-aware evaluation of HAMPER$_{AA}$ together with LID and KD−BU on SVHN, CIFAR10 and CIFAR100. In Table 6.1, we group the results from Table B.1 (that we relegate to Section B.3 due to space constriction) according to the attack-norm (e.g., L_1, L_2, L_∞, No norm), and we express them in terms of the mean on the AUROC↑ and the mean on the FPR$\downarrow_{95\%}$. We also report the average performances per dataset (*Average*) and over the datasets (*Global*).

In general, HAMPER$_{AA}$ outperforms the SOTA detectors by maintaining performance close to 100% AUROC↑ and 0% FPR$\downarrow_{95\%}$ on all the four datasets regardless of the attack-norm considered.

Performance analysis per ε. Overall, the results in Table B.1 show that the smaller the perturbation magnitude ε to craft the attack is, the more complex the attack detection. For example, the worst result of LID is with PGD1 and ε = 15 for SVHN and PGD1 with ε = 20 for CIFAR10 (AUROCs smaller than 50%). KD−BU exhibits the same attitude but for FGSM on SVHN, where it reaches its minimum (5.9% AUROC↑ and 99.5% FPR$\downarrow_{95\%}$) with ε = 0.5. Note that, the high value of the standard deviation in Table 6.1 could implies the detector is more susceptible to the ε changes. This is particularly true in the case of the L_1-norm group since all the attacks considered are created with the same algorithm (PGD). On this regard, KD−BU turns out to be the detector most susceptible (e.g., on CIFAR10 its standard deviation on the FPR$\downarrow_{95\%}$ is 33.4 whilst the one of LID is 8.6 and the one of HAMPER$_{AA}$ is 0.0).

Performance analysis per type of threat. On average between the norm based

Table 6.2: Blind-to-Attack detector performances on the three considered datasets - SVHN, CIFAR10, and CIFAR100 - of the $\texttt{HAMPER}_{\texttt{BA}}$ detector together with the results of the state-of-the-art detection methods, i.e., NSS, averaged over the L_p-norm constraint, along with the average and global performances. The best results among the detectors are shown in **bold**. The results are presented as AUROC↑% $_{\pm \text{FPR}\downarrow_{95\%}\ \%}$ and The results are presented as AUROC↑ $_{\pm \text{FPR}\downarrow_{95\%}\ \%}$ and in terms of mean (μ) and standard deviation (σ).

		NSS			$\texttt{HAMPER}_{\texttt{BA}}$		
		SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100
Norm L_1	μ	69.2 ±78.6	66.7 ±80.1	69.7 ±78.2	**94.9** ±24.9	**94.7** ±22.8	**99.9** ±0.2
	σ	15.7 ±24.3	10.3 ±12.0	16.3 ±32.0	**3.1** ±11.7	**4.7** ±19.6	**0.0** ±0.1
Norm L_2	μ	71.5 ±65.8	68.0 ±72.8	62.5 ±90.2	**94.2** ±22.8	**92.3** ±30.2	**99.9** ±0.3
	σ	19.5 ±36.7	15.7 ±26.8	9.4 ±6.3	**3.4** ±12.6	**7.2** ±23.9	**0.1** ±0.2
Norm L_∞	μ	93.6 ±28.9	92.3 ±15.5	69.1 ±82.0	**98.4** ±7.3	**98.7** ±6.3	**99.9** ±0.3
	σ	10.4 ±43.5	20.1 ±29.2	12.4 ±13.6	**2.4** ±9.8	**2.3** ±10.7	**0.0** ±0.2
No Norm	μ	**99.8** ±0.4	**93.8** ±20.2	92.9 ±24.7	98.5 ±6.4	80.3 ±57.1	**100** ±0.0
	σ	**0.0** ±0.0	**0.0** ±0.0	**0.0** ±0.0	**0.0** ±0.0	**0.0** ±0.0	**0.0** ±0.0
Average	μ	83.5 ±46.3	81.2 ±42.3	68.1 ±81.9	**96.6** ±14.6	**95.8** ±17.0	**99.9** ±0.3
	σ	18.4 ±43.6	21.2 ±39.2	13.4 ±20.1	**3.4** ±13.6	**5.9** ±20.9	**0.1** ±0.2
Global	μ	77.6 ±56.8			**97.4** ±10.6		
	σ	19.2 ±40.0			**4.3** ±16.2		

attacks, LID and KD−BU more easily detects the L_∞-norm attacks. Interestingly, with L_∞ both LID and KD−BU have the best performance on CIFAR10 whilst on SVHN and CIFAR100 the detectors have the best performance in the no norm case (cf. Table 6.1). Consistently over the datasets, KD−BU poorly behaves on CW_2 as the AUROCs do not reach 50% (cf. Table B.1).

Summary. Table 6.1 suggests LID and KD−BU have similar behaviors on SVHN, whilst KD−BU improves on CIFAR10. A closer look at Table B.1 also suggests KD−BU has higher variance in the results w.r.t. LID on all the three datasets. Thus KD−BU performances are most affected by the perturbation magnitude to craft the adversarial examples. $\texttt{HAMPER}_{\texttt{AA}}$, on the other side, is hereby confirmed as the best detector since it does not change its performances no matter the perturbation or the norm considered in the attacks.

Blind-to-attack scenario

Here, we study the performance of the different detectors in the blind-to-attack scenario and show that $\texttt{HAMPER}_{\texttt{BA}}$ outperforms existing detectors.

Global analysis. We present the blind-to-attack evaluation of $\texttt{HAMPER}_{\texttt{BA}}$ together with NSS on SVHN, CIFAR10, CIFAR100. As in Section 6.5.2, in Table 6.2, we group the results from Table B.2 according to the attack-norm and we express them in terms of mean / standard deviation on the AUROC↑ and mean / standard deviation on the $\text{FPR}\downarrow_{95\%}$. Moreover we report the average performances per dataset (*Average*) and over all the dataset (*Global*). On average, $\texttt{HAMPER}_{\texttt{BA}}$ outperforms NSS by 13.1 (31.7) on

SVHN, 14.6(25.3) on CIFAR10 and 31.8(81.6) on CIFAR100 in terms of AUROC↑(FPR↓95%). Under L_1, L_2 and L_∞-norm constraints, $\texttt{HAMPER}_{\texttt{BA}}$ outperforms NSS on all considered datasets. The increase goes up to 30.2(78.0) in L_1-norm, 37.4(89.9) in L_2-norm and 30.8(81.7) in L_∞-norm in terms of in AUROC↑(FPR↓95%).

Performance analysis per ε. Overall, the results in Table B.2 show that, in contrast to $\texttt{HAMPER}_{\texttt{BA}}$, NSS' performances are increasing with the value of maximal allowed perturbation ε. As a matter of fact, the performances of $\texttt{HAMPER}_{\texttt{BA}}$ on L_1 and L_2-norm constraints first decrease as ε increases ($\varepsilon \in [5, 15]$ for L_1-norm constraint, and $\varepsilon \in [0.125, 0.3125]$ for L_2-norm constraint), until it starts increasing. On L_∞-norm constraint, our method follows the expected behavior, i.e., the performances increases with ε.

Performance analysis per type of threat. Table 6.2 suggests that both $\texttt{HAMPER}_{\texttt{BA}}$ and NSS are globally better at detecting attacks with L_∞-norm constraints, particularly those created with PGD and BIM as an attack strategy. However, on SVHN NSS finds more difficult to spot the FGSM-based attacks. SA and DeepFool threats are the thoughest to detect for NSS. On the contrary, while $\texttt{HAMPER}_{\texttt{BA}}$ consistently detect SA-based attack, it shows a slight drop in performance for DeepFool-based attacks on CIFAR10. Finally, NSS presents poor performances at detecting CW_2 attacks, while it is not the case for our proposed method.

Summary. While NSS's performances vary with the dataset, the ε and the norm used to construct the attacks, $\texttt{HAMPER}_{\texttt{BA}}$ consistently detect them. In particular, we note that $\texttt{HAMPER}_{\texttt{BA}}$ is well suited to larger datasets (e.g., CIFAR100). Conversely, NSS' performance highly decreases when passing from the datasets with 10 classes to the one with 100.

6.5.3 Attacking HAMPER using adaptive adversary

The importance of attacking defenses with adaptive attacks has increased recently [ABUSNAINA and collab., 2021; RAGHURAM and collab., 2021]. As mentionned in Subsection 6.2.3, the Backward-Pass Differentiable Attack (*BPDA*; ATHALYE and collab., 2018a) is based on the possibility to find a suitable surrogate to the non-differentiable parts of any defense. However, deriving a suitable differentiable surrogate of the halfspace-mass depth remains a open research question which has never been tackled. As a matter of fact, the only attempts to approximate a non-differentiable depth was performed on the Tuckey depth in SHE and collab. [2021], with very poor results [DYCKERHOFF and collab., 2021]. Finding a differentiable suitable surrogate to attack HAMPER would, therefore, require a substantial effort and should be rigorously handled. As a consequence, we have to rely on adaptive blackbox attackers, as suggested in TRAMER and collab. [2020], to attack HAMPER.

Experimental Setting. We designed a blackbox adaptive attacker by using SA,

Table 6.3: Detector performances and Attack's success under adaptive blackbox attacks for NSS and $\text{HAMPER}_{\text{BA}}$ on CIFAR10. We present the results as: AUROC↑% $_{\pm\text{FPR}\downarrow_{95\%}\ \%}$ for the detector performances.

Adaptive Attacks				
CIFAR10				
	NSS		$\text{HAMPER}_{\text{BA}}$	
	Detector Performances	Attack Success (%)	Detector Performances	Attack Success (%)
$\alpha = 10^{-3}$	9.4 $_{\pm 100}$	95.82	79.3 $_{\pm 67.5}$	98.93
$\alpha = 10^{-2}$	7.2 $_{\pm 100}$	96.12	95.4 $_{\pm 26.5}$	99.36
$\alpha = 10^{-1}$	5.5 $_{\pm 100}$	93.60	74.6 $_{\pm 65.8}$	99.32
$\alpha = 1$	7.8 $_{\pm 100}$	95.88	85.5 $_{\pm 66.1}$	98.77
$\alpha = 10^{1}$	3.2 $_{\pm 100}$	93.28	63.6 $_{\pm 87.1}$	95.83
$\alpha = 10^{2}$	3.5 $_{\pm 100}$	93.83	68.0 $_{\pm 75.6}$	42.42
$\alpha = 10^{3}$	3.7 $_{\pm 100}$	93.44	47.7 $_{\pm 86.1}$	25.06

which is both effective and computationally efficient [ENGSTROM and collab., 2019]. To extend SA into an adaptive attack, we modified the success criterion to not only fool the classifier but also the detector, and the loss, which becomes a trade-off between minimizing the difference between the logits of the two most probable classes and maximizing the detector's prediction. In Table 6.3, we present the results of the adaptive SA attack on CIFAR10 for both NSS and $\text{HAMPER}_{\text{BA}}$.

Analysis. On Table 6.3, we varied the value of the parameter α controlling the trade-off between the classifier and the detector performances. As α increases, the performance of the attack on the classifier decreases while the detector has more and more trouble detecting them, as one can expect. On the considered adaptive attacks, it is clear that it is easy for the attacker to find powerful and undetected samples to attack NSS. However, it is more difficult to fool our proposed method. To decrease the AUROC↑ to a value below 50 (which is equivalent to a random detector), the attacker is only able to fool the classifier 25% of the time.

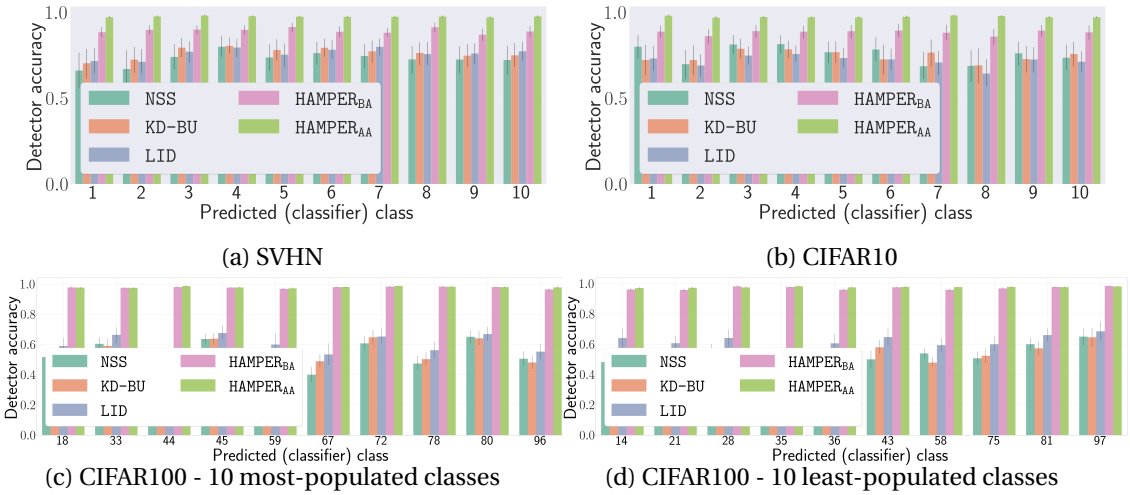
Takeaways. $\text{HAMPER}_{\text{BA}}$ is more robust to adaptive attacks than NSS.

6.5.4 Further analysis of the detector behaviors

In this section, we first investigate the per-class behavior of the detectors to further asses the effectiveness of the proposed method (cf. Section 6.5.4). Then, we study the AUROC↑/$\text{FPR}\downarrow_{95\%}$ trade-off (cf. Section 6.5.4) and the time and resource constraints (cf. Section 6.5.4) for each of the considered methods.

Analyzing the detectors' per-class behavior

Subsection 6.4.2 identifies a class-dependant behavior of the attack mechanisms that could translate in a class-dependant behavior of the different detection methods. In



(a) SVHN (b) CIFAR10

(c) CIFAR100 - 10 most-populated classes (d) CIFAR100 - 10 least-populated classes

Figure 6.5: **Accuracies of the detectors per predicted classifier class**. For visualization reasons, we restrict the plot to the 10-most/least populated classes for CIFAR100.

this experiment, we study the performance distribution per class.

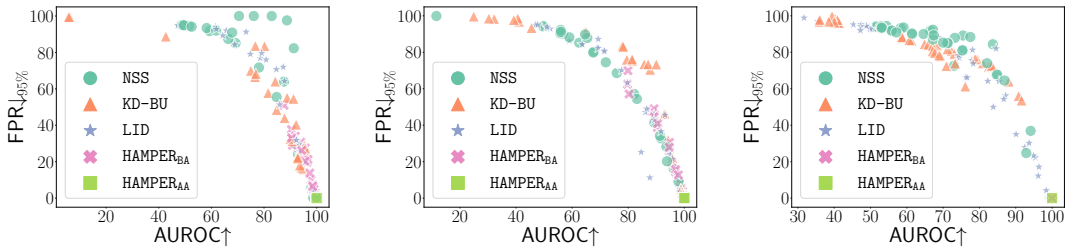
Simulation. In Figure 6.5, we examine the accuracy of the detectors on the testing samples (natural and adversarial) w.r.t. the class predicted by the classifier. Note that, for visualization purposes, we select the 10-most/least populated classes in the case of CIFAR100. For this simulation, we select the threshold γ for which TPR is at 95%.

Analysis. HAMPER performances are not affected either by the class nor by the dataset and they consistently outperform the competitors. However, the competitor's performance are class-dependant: they tend to better distinguish between natural and attacked samples for classes with the highest number of adversarial examples. This is demonstrated by Figure 6.5a where NSS, KD-BU and LID obtain the highest accuracy in class 4 on SVHN, which is also the most populated class (cf. Figure 6.2a). A similar behavior is observable in CIFAR10 (cf. classes 3 and 4 in Figure 6.5b and Figure 6.2b). Moreover, Figure 6.5 suggests that, in terms of accuracy per class, the SOTA methods show similar performances on SVHN; on the contrary, on CIFAR10 the detector performing the best is NSS while on CIFAR100 it is LID.

Takeaways. *Differently from the competitors, the HAMPER detectors are not affected by the per-class distribution of the samples.* In particular, and regardless of the dataset, the proposed detectors show a uniform behavior overall the classes. Further confirmation is given from the plots in Figure 6.5c and Figure 6.5d where the accuracies of the detectors remain constant even when focusing on only the most and the least-populated classes respectively.

Studying the AUROC↑-FPR$\downarrow_{95\%}$ relationship

As commonly done in anomaly detection, we measure the detection performances in terms of AUROC↑ and FPR$\downarrow_{95\%}$. However, to a large AUROC↑ does not necessarily



(a) SVHN (b) CIFAR10 (c) CIFAR100

Figure 6.6: FPR$\downarrow_{95\%}$ as a function of the AUROC↑ for all five considered methods, (a) on SVHN, (b) on CIFAR10, and (c) on CIFAR100.

correspond a low FPR$\downarrow_{95\%}$. In this experiment, we study the trade-off between AUROC↑ and FPR$\downarrow_{95\%}$ performances for each considered detection method.

Simulation. In Figure 7.5, we analyze the trade-off between AUROC↑ and FPR$\downarrow_{95\%}$. We translate on the Cartesian planes the results presented in Table B.1 and in Table B.2. The perfect detector will have the points in (100, 0).

Analysis. Interestingly, the HAMPER detectors behave closely to the perfect detector for all the considered attacks. This is particularly true for HAMPER$_{\text{AA}}$. For NSS, KD-BU and LID, a large AUROC↑ does not necessarily correspond a low FPR$\downarrow_{95\%}$. In SVHN (Figure 6.6a), five of the NSS points are exhibiting high AUROC↑ (between 70% and 92%) while presenting extremely high FPR$\downarrow_{95\%}$ (between 81% and 99.6%). A similar behavior is presented in CIFAR10 with LID.

Takeaways. Contrary to other detection method, which can exhibit high FPR$\downarrow_{95\%}$ for high AUROC↑, our proposed detectors behave similarly to the perfect detector for both attack-aware and blind-to-attack scenarios.

Time and resources constraints

For some applications, time and resource necessity can be critical. We, therefore, decided to measure the constraints of each considered method.

Simulations. In Table 6.4, We report the time and resource constraints needed for each method.

Table 6.4: Time and computational constraints to train and test each detection method.

Method	GPUs	Training Time	Testing Time
NSS	V100-16G	00m30s	00m55s
KD-BU	V100-16G	00m30s	02m00s
LID	V100-16G	04m00s	35m00s
HAMPER$_{\text{AA}}$	V100-16G	02m00s	02m00s
HAMPER$_{\text{BA}}$	V100-16G	02m00s	02m00s

Analysis. All methods have quite comparable training time. However, LID, due to the extraction of the LID parameters to all layers, takes a lot more time to test than the others.

Takeaways. HAMPER's deployment requires comparable time and resources to the other considered detectors.

6.6 Concluding Remarks and Future Work

In this paper, we introduced HAMPER, a simple and effective method to detect adversarial attacks. One of the keys of HAMPER is to rely on the halfspace-mass depth, a statistical tool that remains overlooked by the machine learning community. Through our extensive experiments based on two scenarios, attack-aware, and blind-to-attack, we demonstrate that HAMPER achieves state-of-the-art performances. On average, it outperforms the existing best detector by 13.1% (29.0%) in terms of AUROC↑ (resp. $\text{FPR}\downarrow_{95\%}$) on SVHN, 14.6% (25.7%) on CIFAR10, and 22.6% (49.0%) on CIFAR100. Interestingly, HAMPER exhibits class-independence, and less dependence on the norm-attack and the threat scenarios than other adversarial attack detection methods.

Like all the supervised adversarial detection methods in the literature, HAMPER requires knowledge of the kind of attack evaluated at testing time, meaning that they are generally validated by assuming a single implicitly known attack strategy, which does not necessarily account for real-life threats. In future work, we will investigate how HAMPER and its main competitors perform when facing samples crafted through multiple unknown attack strategies at test time. We will mainly make an effort to understand whether the notion of depth is well suited to better generalize the detection of adversarial examples to attacks that are not involved in the supervised framework.

Broader Impact Statement

Many concerns have been raised about the potential failures of Deep Learning: large neural networks are not trustworthy enough, limiting their adoption in high-risk applications. This paper's main contribution aims at improving the reliability of Deep Learning by designing a tool to prevent a malicious agent from disrupting the functioning of the system. Thus we believe our work will have a positive impact on society.

Chapter 6 Conclusion

In this work, we presented an efficient supervised detection method designed to defend DNNs against adversarial attacks. Given that attackers have different behavior with respect to the predicted class, we leverage this knowledge by constructing a class-wise detection method. In addition, we observed that using only the output of the final layers of the network to protect was enabling us to design an efficient detection method. We experimentally proved its efficiency compared to other state-of-the-art method on two scenario: attack-aware and blind-to-attack. Finally, we showed that our method was toughening the adaptive attacker's goal to fool both the network and the defense, compared to other detection methods.
In this work, we set ourselves in the supervised setting. In what follows, we will present our proposed framework to detect adversarial examples in the unsupervised case, focusing on protecting a widely overlooked classifier architecture: the Vision Transformers.

6.7 References

ABUSNAINA, A., Y. WU, S. ARORA, Y. WANG, F. WANG, H. YANG and D. MOHAISEN. 2021, «Adversarial example detection using latent neighborhood graph», in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, p. 7687–7696. 163, 171, 178

ALDAHDOOH, A., W. HAMIDOUCHE and O. DÉFORGES. 2021a, «Revisiting model's uncertainty and confidences for adversarial example detection», *arXiv preprint arXiv: 2103.05354*. 164

ALDAHDOOH, A., W. HAMIDOUCHE, S. A. FEZZA and O. DÉFORGES. 2021b, «Adversarial example detection for DNN models: A review», *arXiv preprint arXiv:2105.00203*. 163

ANDRIUSHCHENKO, M., F. CROCE, N. FLAMMARION and M. HEIN. 2020, «Square attack: a query-efficient black-box adversarial attack via random search», in *European Conference on Computer Vision*, Springer, p. 484–501. 167

ATHALYE, A., N. CARLINI and D. WAGNER. 2018a, «Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples», in *International conference on machine learning*, PMLR, p. 274–283. 167, 168, 178

ATHALYE, A., N. CARLINI and D. A. WAGNER. 2018b, «Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples», in *Proceedings*

of the 35th International Conference on Machine Learning, ICML, vol. 80, PMLR, p. 274–283. 163

BREUNIG, M. M., H.-P. KRIEGEL, R. T. NG and J. SANDER. 2000, «Lof: identifying density-based local outliers», in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, p. 93–104. 164

CARLINI, N. and D. WAGNER. 2017, «Towards evaluating the robustness of neural networks», in *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, p. 39–57. 167

CARRARA, F., R. BECARELLI, R. CALDELLI, F. FALCHI and G. AMATO. 2018, «Adversarial examples detection in features distance spaces», in *Computer Vision - ECCV 2018 Workshops - Munich, Germany, Proceedings, Part II*, vol. 11130, Springer, p. 313–327. 163

CHANDOLA, V., A. BANERJEE and V. KUMAR. 2009, «Anomaly detection: A survey», *ACM Comput. Surv.*, vol. 41, nº 3, p. 15:1–15:58, ISSN 0360-0300. 164

CHEN, B., K. M. TING, T. WASHIO and G. HAFFARI. 2015, «Half-space mass: a maximally robust and efficient data depth method», *Machine Learning*, vol. 100, nº 2, p. 677–699. 168, 169, 170

CHEN, J., M. I. JORDAN and M. J. WAINWRIGHT. 2020, «Hopskipjumpattack: A query-efficient decision-based attack», in *2020 ieee symposium on security and privacy (sp)*, IEEE, p. 1277–1294. 167

CHEN, Y., X. DANG, H. PENG and H. BART. 2009, «Outlier detection with the kernelized spatial depth function», *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, nº 2, p. 288–305. 164

CROCE, F. and M. HEIN. 2020, «Minimally distorted adversarial examples with a fast adaptive boundary attack», in *International Conference on Machine Learning*, PMLR, p. 2196–2205. 163

CUEVAS, A., M. FEBRERO and R. FRAIMAN. 2007, «Robust estimation and classification for functional data via projection-based depth notions», *Computational Statistics*, vol. 22, nº 3, p. 481–496. 164

CUEVAS, A. and R. FRAIMAN. 2009, «On depth measures and dual statistics. a methodology for dealing with general data», *Journal of Multivariate Analysis*, vol. 100, nº 4, p. 753–766. 169

DAVIS, J. and M. GOADRICH. 2006, «The relationship between precision-recall and roc curves», in *Proceedings of the 23rd international conference on Machine learning*, p. 233–240. 175

DYCKERHOFF, R., P. MOZHAROVSKYI and S. NAGY. 2021, «Approximate computation of projection depths», *Computational Statistics & Data Analysis*, vol. 157, p. 107 166. 178

ENGSTROM, L., B. TRAN, D. TSIPRAS, L. SCHMIDT and A. MADRY. 2019, «Exploring the landscape of spatial robustness», in *International Conference on Machine Learning*, PMLR, p. 1802–1811. 167, 179

FEINMAN, R., R. R. CURTIN, S. SHINTRE and A. B. GARDNER. 2017, «Detecting adversarial samples from artifacts», *arXiv preprint arXiv:1703.00410*. 163, 164, 166, 170

GEIFMAN, Y. and R. EL-YANIV. 2019, «Selectivenet: A deep neural network with an integrated reject option», in *Proceedings of the 36th International Conference on Machine Learning, ICML*, vol. 97, p. 2151–2159. 163

GEIFMAN, Y., G. UZIEL and R. EL-YANIV. 2019, «Reduced uncertainty estimation for deep neural classifiers», in *7th International Conference on Learning Representations, ICLR*. 163

GOODFELLOW, I. J., J. SHLENS and C. SZEGEDY. 2015, «Explaining and harnessing adversarial examples», *International Conference on Learning Representations*. 167

GROSSE, K., P. MANOHARAN, N. PAPERNOT, M. BACKES and P. D. MCDANIEL. 2017, «On the (statistical) detection of adversarial examples», *arXiv preprint arXiv:1702.06280*. 163

GUO, C., G. PLEISS, Y. SUN and K. Q. WEINBERGER. 2017, «On calibration of modern neural networks», in *Proceedings of the 34th International Conference on Machine Learning, ICML*, vol. 70, p. 1321–1330. 163

HASTIE, T., R. TIBSHIRANI, J. H. FRIEDMAN and J. H. FRIEDMAN. 2009, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2, Springer. 174

HU, S., T. YU, C. GUO, W.-L. CHAO and K. Q. WEINBERGER. 2019, «A new defense against adversarial images: Turning a weakness into a strength», *Advances in Neural Information Processing Systems*, vol. 32. 163

JÖRNSTEN, R. 2004, «Clustering and classification based on the l1 data depth», *Journal of Multivariate Analysis*, vol. 90, nº 1, p. 67–89. 164

KHERCHOUCHE, A., S. A. FEZZA, W. HAMIDOUCHE and O. DÉFORGES. 2020, «Natural scene statistics for detecting adversarial examples in deep neural networks», in *22nd IEEE International Workshop on Multimedia Signal Processing*, IEEE, p. 1–6. 163, 166, 170

KRIZHEVSKY, A. 2009, «Learning multiple layers of features from tiny images», cahier de recherche. 172

KURAKIN, A., I. J. GOODFELLOW and S. BENGIO. 2018, «Adversarial examples in the physical world», in *Artificial intelligence safety and security*, Chapman and Hall/CRC, p. 99–112. 167

LANGE, T., K. MOSLER and P. MOZHAROVSKYI. 2014, «Fast nonparametric classification based on data depth», *Statistical Papers*, vol. 55, nº 1, p. 49–69. 164

LI, X. and F. LI. 2017, «Adversarial examples detection in deep networks with convolutional filter statistics», in *IEEE International Conference on Computer Vision, ICCV*, IEEE Computer Society, p. 5775–5783. 163

LIANG, B., H. LI, M. SU, X. LI, W. SHI and X. WANG. 2021, «Detecting adversarial image examples in deep neural networks with adaptive noise reduction», *IEEE Trans. Dependable Secur. Comput.*, vol. 18, nº 1, p. 72–85. 163

LIU, F. T., K. M. TING and Z.-H. ZHOU. 2008, «Isolation forest», in *2008 eighth ieee international conference on data mining*, IEEE, p. 413–422. 164

LU, J., T. ISSARANON and D. A. FORSYTH. 2017, «Safetynet: Detecting and rejecting adversarial examples robustly», in *IEEE International Conference on Computer Vision*, IEEE Computer Society, p. 446–454. 163

MA, S., Y. LIU, G. TAO, W. LEE and X. ZHANG. 2019, «NIC: detecting adversarial samples with neural network invariant checking», in *26th Annual Network and Distributed System Security Symposium*, The Internet Society. 163

MA, X., B. LI, Y. WANG, S. M. ERFANI, S. N. R. WIJEWICKREMA, G. SCHOENEBECK, D. SONG, M. E. HOULE and J. BAILEY. 2018, «Characterizing adversarial subspaces using local intrinsic dimensionality», in *6th International Conference on Learning Representations*. 163, 166, 170

MADRY, A., A. MAKELOV, L. SCHMIDT, D. TSIPRAS and A. VLADU. 2018, «Towards deep learning models resistant to adversarial attacks», in *International Conference on Learning Representations*. 163, 167

MEINKE, A. and M. HEIN. 2020, «Neural networks that provably know when they don't know», in *8th International Conference on Learning Representations, ICLR*. 163

MENG, D. and H. CHEN. 2017, «Magnet: A two-pronged defense against adversarial examples», in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, édité par B. M. Thuraisingham, D. Evans, T. Malkin and D. Xu, ACM, p. 135–147. 163

METZEN, J. H., T. GENEWEIN, V. FISCHER and B. BISCHOFF. 2017, «On detecting adversarial perturbations», in *5th International Conference on Learning Representations*. 163

MOOSAVI-DEZFOOLI, S.-M., A. FAWZI and P. FROSSARD. 2016, «Deepfool: a simple and accurate method to fool deep neural networks», in *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 2574–2582. 167

NETZER, Y., T. WANG, A. COATES, A. BISSACCO, B. WU and A. Y. NG. 2011, «Reading digits in natural images with unsupervised feature learning», in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. 172

OJA, H. 1983, «Descriptive statistics for multivariate distributions», *Statistics & Probability Letters*, vol. 1, nº 6, p. 327 – 332. 164

PANG, T., C. DU, Y. DONG and J. ZHU. 2018, «Towards robust detection of adversarial examples», in *Advances in Neural Information Processing Systems 31*, p. 4584–4594. 164

PICOT, M., F. MESSINA, M. BOUDIAF, F. LABEAU, I. B. AYED and P. PIANTANIDA. 2021, «Adversarial robustness via fisher-rao regularization», *arXiv preprint arXiv: 2106.06685*. 163

RAGHURAM, J., V. CHANDRASEKARAN, S. JHA and S. BANERJEE. 2021, «A general framework for detecting anomalous inputs to dnn classifiers», in *International Conference on Machine Learning*, PMLR, p. 8764–8775. 178

RAMSAY, K., S. DUROCHER and A. LEBLANC. 2019, «Integrated rank-weighted depth», *Journal of Multivariate Analysis*, vol. 173, p. 51–69. 168

SCHÖLKOPF, B., J. C. PLATT, J. SHAWE-TAYLOR, A. J. SMOLA and R. C. WILLIAMSON. 2001, «Estimating the support of a high-dimensional distribution», *Neural computation*, vol. 13, nº 7, p. 1443–1471. 164

SERFLING, R. 2006, «Depth functions in nonparametric multivariate inference», *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 72, p. 1. 164

SHE, Y., S. TANG and J. LIU. 2021, «On generalization and computation of tukey's depth: Part i», *arXiv preprint arXiv:2112.08475*. 178

SOTGIU, A., A. DEMONTIS, M. MELIS, B. BIGGIO, G. FUMERA, X. FENG and F. ROLI. 2020, «Deep neural rejection against adversarial examples», *EURASIP J. Inf. Secur.*, vol. 2020, p. 5. 164

STAERMAN, G. 2022, *Functional anomaly detection and robust estimation*, thèse de doctorat, Institut polytechnique de Paris. 170

STAERMAN, G., P. MOZHAROVSKYI, S. CLÉMENÇON and F. D'ALCHÉ BUC. 2019, «Functional isolation forest», in *Proceedings of The Eleventh Asian Conference on Machine Learning*, vol. 101, p. 332–347. 164

STAERMAN, G., P. MOZHAROVSKYI and S. CLÉMENÇON. 2020, «The area of the convex hull of sampled curves: a robust functional statistical depth measure», in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, vol. 108, PMLR, p. 570–579. 164

STAERMAN, G., P. MOZHAROVSKYI, S. CLÉMENÇON and F. D'ALCHÉ BUC. 2021a, «A pseudo-metric between probability distributions based on depth-trimmed regions», *arXiv preprint arXiv:2103.12711*. 164

STAERMAN, G., P. MOZHAROVSKYI and S. CLÉMENÇON. 2021b, «Affine-invariant integrated rank-weighted depth: Definition, properties and finite sample analysis», *arXiv preprint arXiv:2106.11068*. 164, 168

SZEGEDY, C., W. ZAREMBA, I. SUTSKEVER, J. BRUNA, D. ERHAN, I. J. GOODFELLOW and R. FERGUS. 2014, «Intriguing properties of neural networks», in *2nd International Conference on Learning Representations*. 162, 166

TAO, G., S. MA, Y. LIU and X. ZHANG. 2018, «Attacks meet interpretability: Attribute-steered detection of adversarial samples», in *Advances in Neural Information Processing Systems 31*, p. 7728–7739. 164

TRAMER, F., N. CARLINI, W. BRENDEL and A. MADRY. 2020, «On adaptive attacks to adversarial example defenses», *Advances in Neural Information Processing Systems*, vol. 33, p. 1633–1645. 167, 178

TUKEY, J. W. 1975, «Mathematics and the picturing of data», in *Proceedings of the International Congress of Mathematicians*, vol. 2, p. 523–531. 164, 168

XU, W., D. EVANS and Y. QI. 2018, «Feature squeezing: Detecting adversarial examples in deep neural networks», in *25th Annual Network and Distributed System Security Symposium*, The Internet Society. 163

YAO, C., P. BIELIK, P. TSANKOV and M. VECHEV. 2021, «Automated discovery of adaptive attacks on adversarial defenses», *Advances in Neural Information Processing Systems*, vol. 34, p. 26 858–26 870. 167

ZHENG, S., Y. SONG, T. LEUNG and I. J. GOODFELLOW. 2016, «Improving the robustness of deep neural networks via stability training», in *2016 IEEE Conference*

on Computer Vision and Pattern Recognition, CVPR, IEEE Computer Society, p. 4480–4488. 163

ZHENG, T., C. CHEN and K. REN. 2019, «Distributionally adversarial attack», in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, AAAI Press, p. 2253–2260. 163

ZHENG, Z. and P. HONG. 2018, «Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks», in *Advances in Neural Information Processing Systems 31*, p. 7924–7933. 164

ZUO, Y. and R. SERFLING. 2000, «General notions of statistical depth function», *The Annals of Statistics*, vol. 28, nº 2, p. 461–482. 164, 170

Chapter 7

A Simple Unsupervised Data Depth-based Method to Detect Adversarial Images

Chapter 7 Abstract

We here present our final contribution. In the previous chapter, we made use of data-depths to detect adversarial examples in a supervised setting. In this work, we wanted to free our method from the necessity to gather knowledge about the possible attacker. In addition, we observed that, no other previous work focused on developing detection methods on Vision Transformer, in spite of it being the SOTA architecture to classify images. We therefore propose a detection method that leverages the particular structure of Vision Transformers, and the data-depths (more precisely the Integrated-Rank-Weighted depth) to efficiently detect adversarial examples. We experimentally show the advantages of our method compared to previous state-of-the-art unsupervised method. Moreover, we show that directly attacking our proposed detector is far from being straightforward and that our method makes the adaptive attacker's job more complicated.

Contents

Abstract

Deep neural networks suffer from critical vulnerabilities regarding robustness, which limits their exploitation in many real-world applications. In particular, a serious concern is their inability to defend against adversarial attacks. Although the research community has developed a large amount of effective attacks, the detection problem has received little attention. Existing detection methods either rely on additional training or on specific heuristics at the risk of overfitting. Moreover, they have mainly focused on ResNet architectures while transformers, which are state-of-the-art for vision tasks, have not been properly investigated. In this paper, we overcome these limitations by introducing APPROVED, a simple unsupervised detection method for transformer architectures. It leverages the information available in the logit layer and computes a similarity score with respect to the training distribution. This is accomplished using a *data depth* that is: *(i)* computationally efficient; and *(ii)* non-differentiable, making it harder for gradient-based adversaries to craft malicious samples. Our extensive experiments show that APPROVED consistently outperforms previous detectors on CIFAR10, CIFAR100 and Tiny ImageNet.

7.1 Introduction

Recent years have seen a rapid development of Deep Neural Networks (DNNs), which have led to a significant improvement over previous state-of-the-art methods (SOTA) in numerous decision-making tasks. However, together with this growth, concerns have been raised about the potential failures of deep learning systems, which limit their large-scale adoption [ALVES and collab., 2018; JOHNSON, 2018; SUBBASWAMY and

SARIA, 2020]. In Computer Vision, a particular source of concern is the existence of *adversarial attacks* [SZEGEDY and collab., 2014], which are samples created by adding to the original (clean) image a well-designed additive perturbation, imperceptible to human eyes, with the goal of fooling a given classifier. The vulnerability of DNNs to such kinds of attacks limits their deployment in safety-critical systems as in aviation safety management [ALI and collab., 2020], health monitoring systems [LEIBIG and collab., 2017; MEINKE and HEIN, 2020]) or in autonomous driving [BOJARSKI and collab., 2016; GUO and collab., 2017]. Therefore, it is crucial to deploy a proper strategy to defend against adversarial attacks [AMODEI and collab., 2016].

In this context, the task of distinguishing adversarial samples from clean ones is becoming increasingly challenging as developing new attacks is getting more attention from the community [CROCE and HEIN, 2020; DENG and KARAM, 2020a,b; DONG and collab., 2019; DUAN and collab., 2020; GAO and collab., 2021; JIA and collab., 2020; LIN and collab., 2019; NASEER and collab., 2021; WANG and collab., 2021a; WU and collab., 2020b; ZHAO and collab., 2020]. Inspired by the concept of *rejection channels* [CHOW, 1957], which was proposed over 60 years ago for the character recognition problem, one way to address adversarial attacks is to construct a detector-based rejection strategy. Its objective is to discriminate malicious samples from clean ones, which implies discarding samples detected as adversarial. Research in this area focuses on both *supervised* and *unsupervised* approaches [ALDAHDOOH and collab., 2021c]. The supervised approaches rely on features extracted from adversarial examples generated according to one or more attacks [FEINMAN and collab., 2017; KHERCHOUCHE and collab., 2020; MA and collab., 2018]; the unsupervised ones, instead, do not rely on prior knowledge of attacks and, in general, only learn from clean data at the time of training [MENG and CHEN, 2017; XU and collab., 2018].

In this work, we focus on the unsupervised scenario, which is often a reasonable approach to real-world use-cases. We model the adversarial detection problem as an *anomaly detection* framework [BREUNIG and collab., 2000; CHANDOLA and collab., 2009; LIU and collab., 2008; SCHÖLKOPF and collab., 2001; STAERMAN and collab., 2019, 2020], where the aim is to identify abnormal observations without seeing them during training. Statistical tools called *data depths* are natural similarity score in this context. Data depths have a simple geometric interpretation as they provide center-outward ordering of points with respect to a probability distribution [TUKEY, 1975; ZUO and SERFLING, 2000]. Geometrically speaking, the data depths measure how deep a sample is in a given distribution. Although data depths have received attention from the statistical community, they remain overlooked by the machine learning community.

Contributions. Our contributions can be summarized as follows:

1. **Building on novel tools: data depths.** Our first contribution is to introduce `AP-`

APPROVED, A simPle unsuPeRvised method fOr adVersarial imagE Detection. Given an input, APPROVED considers its embedding in the last layer of the pre-trained classifier and computes the depth of the sample w.r.t the training probability distribution. The deeper it is, the less likely it is to be adversarial. Contrarily to existing methods that involve additional networks training [MENG and CHEN, 2017] or heavily rely on opaque feature engineering [XU and collab., 2018], APPROVED is computationally efficient and has a simple geometrical interpretation. Moreover, data depths non-differentiability making it harder for to gradient-based attackers to target APPROVED.

2. **A truly upgraded experimental setting.** Motivated by practical considerations which are different from previous works [FEINMAN and collab., 2017; KHERCHOUCHE and collab., 2020; MA and collab., 2018; MENG and CHEN, 2017; XU and collab., 2018] focusing on ResNets [HE and collab., 2016], we choose to benchmark APPROVED on vision transformers models [CHEN and collab., 2021; DOSOVITSKIY and collab., 2021; STEINER and collab., 2021; TOLSTIKHIN and collab., 2021; ZHAI and collab., 2022]. Indeed, such networks achieve state-of-the-art results on several visual tasks, including object detection [HE and collab., 2021], image classification [WANG and collab., 2021b] and generation [PARMAR and collab., 2018], largely outperforming ResNets. Interestingly enough, we empirically observe that transformers behave differently from ResNets, which justifies the need to develop detection techniques such as APPROVED, that leverage the specific features of transformers' architectures. Moreover, to avoid overfitting on a specific attack, we test our detection method on a wide range of attack mechanisms.

3. **Ensuring reproducibility.** We provide the open-source code of our method, attacks, and baseline to ensure reproducibility and reduce future research computation and coding overhead.

Organization of the paper. The paper is organized as follows. In Section 7.2, we review detection methods along with attack mechanisms. In Section 7.3, we introduce our detector APPROVED and focus on the description of the data depth on which it relies. In Section 7.4, we study the performance of adversarial attacks on vision transformers and give insights on models' behavior under threat. In Section 7.5, we evaluate APPROVED through numerical experiments and concluding remarks are relegated to Section 7.6.

7.2 Background and Related Work

Notations. Let us consider the classical supervised learning problem where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ denotes the input sample in the space $\mathcal{X}$, and $y \in \mathcal{Y} = \{1, \dots, C\}$ denotes its associated label. The unknown data distribution is denoted by $p_{\mathbf{XY}}$. The training

dataset $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is defined as $n \geq 1$ independent identically distributed (i.i.d.) realizations of $p_{\mathbf{XY}}$. Consider $\mathscr{D}_c = \{(\mathbf{x}_i, y_i) \in \mathscr{D} : y_i = c\}$, the training data for a given class $c \in \mathscr{Y}$. We define the empirical training distribution for the class c at layer ℓ as $p_c^\ell = \frac{1}{|\mathscr{D}_c|} \sum_{\mathbf{x} \in \mathscr{D}_c} \delta_{f_{\boldsymbol{\theta}}^\ell(\mathbf{x})}$ where δ_u is the dirac mass at point u. Let $f_{\boldsymbol{\theta}}^\ell : \mathscr{X} \to \mathbb{R}^{d_\ell}$ with $\ell \in \{1, \ldots, \mathrm{L}\}$, denote the output of the ℓ-th layer of the DNN, where d_ℓ is the dimension of the latent space induced by the ℓ-th layer. The class prediction is obtained from the L-th layer softmax output as follows:

$$f_{\boldsymbol{\theta}}^{\mathrm{L}}(\mathbf{x}) \triangleq \underset{c \in \mathscr{Y}}{\arg\max}\, q_{\boldsymbol{\theta}}(c|\mathbf{x}) \text{ with } q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}) = \mathrm{softmax}(f_{\boldsymbol{\theta}}^{\mathrm{L}-1}(\mathbf{x})).$$

7.2.1 Review of attack mechanisms

The existence of adversarial examples and their capability to lure a deep neural network have been first introduced in Szegedy and collab. [2014]. The authors define the adversarial generation problem as:

$$\mathbf{x}' = \underset{\mathbf{x}' \in \mathbb{R}^d : \|\mathbf{x}' - \mathbf{x}\|_p < \varepsilon}{\arg\min} \|\mathbf{x}' - \mathbf{x}\|_p \text{ s.t. } f_{\boldsymbol{\theta}}^{\mathrm{L}}(\mathbf{x}') \neq y, \tag{7.1}$$

where y is the true label associated to a natural sample $\mathbf{x} \in \mathscr{X}$ being modified, $\|\cdot\|_p$ is the L_p-norm operator, and ε is the maximal allowed perturbation.

Multiple techniques have since been crafted to solve this problem. They can be divided into two main groups of attack mechanisms depending on the knowledge they have of the DNN model: whitebox and blackbox attacks. The former has full access to the model, its weights, and gradients, while the latter can only rely on queries.

Carlini & Wagner's (CW) [Carlini and Wagner, 2017] attack is among the strongest whitebox attacks developed yet. It attempts to solve the adversarial problem in Equation 7.1 by regularizing the minimization of the perturbation norm by a surrogate of the misclassification constraint. *DeepFool (DF)* [Moosavi-Dezfooli and collab., 2016] is an iterative method that solves a locally linearized version of the adversarial problem and takes a step in that direction.

The authors of Goodfellow and collab. [2014] relax the problem as follows:

$$\mathbf{x}' = \underset{\mathbf{x}' \in \mathbb{R}^d : \|\mathbf{x}' - \mathbf{x}\|_p < \varepsilon}{\arg\max} \mathscr{L}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}), \tag{7.2}$$

where $\mathscr{L}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$ is the objective of the attacker, which is a surrogate of the misclassification constraint, and propose the *Fast Gradient Sign Method (FGSM)* that approximates the solution of the relaxed problem in Equation 7.2 by taking one step in the direction of the sign of the gradient of the attacker's objective w.r.t. the input. *Basic Iterative Method (BIM)* [Kurakin and collab., 2018] and *Projected Gradient*

Descent (PGD) [MADRY and collab., 2018] are two iterative extensions of the FGSM algorithm. Their main difference relies on the initialization of the attack algorithm, i.e., while BIM initializes the adversarial examples to the original samples PGD adds a random uniform noise on it. Although created for L_∞-norm constraints, these three methods can be extended to any L_p-norm constraint.

To overcome the absence of knowledge about the model to attack, *Hop Skip Jump (HOP)* [CHEN and collab., 2020] tries to estimate the model's gradient through queries. *Square Attack (SA)* [ANDRIUSHCHENKO and collab., 2020] is based on random searches for a perturbation. If the perturbation doesn't increase the attacker's objective, it is discarded. Finally, *Spatial Transformation Attack (STA)* [ENGSTROM and collab., 2019] rotates and translates the original samples to fool the model.

7.2.2 Review of detection methods

Defending methods against adversarial attacks have been widely studied for classical CNNs [ALAYRAC and collab., 2019; ATZMON and collab., 2019; CARMON and collab., 2019; HENDRYCKS and collab., 2019; HUANG and collab., 2020; MADRY and collab., 2018; RICE and collab., 2020; WANG and collab., 2019; WU and collab., 2020a; ZHANG and collab., 2019]. Whereas a few works have focused on studying the robustness of vision transformers to adversarial samples [ALDAHDOOH and collab., 2021a; BENZ and collab., 2021; MAHMOOD and collab., 2021]. Meanwhile, to protect adversarial attacks from disrupting DNNs' functioning, it is possible to craft detectors to ensure that the sample can be trusted.

Building a detector falls down to finding a scoring function $s : \mathbb{R}^d \to \mathbb{R}$ and a threshold $\gamma \in \mathbb{R}$ to build a binary rule $g : \mathbb{R}^d \to \{0, 1\}$. For a given test sample $\mathbf{x} \in \mathbb{R}^d$,

$$g(\mathbf{x}) = \mathbb{I}\{s(\mathbf{x}) > \gamma\} = \begin{cases} 1 \text{ if } s(\mathbf{x}) > \gamma, \\ 0 \text{ if } s(\mathbf{x}) \leq \gamma. \end{cases} \tag{7.3}$$

If s is an anomaly score, $g(\mathbf{x}) = 0$ implies that $\mathbf{x}$ is considered as 'natural', i.e., sampled from $p_{\mathbf{XY}}$, and $g(\mathbf{x}) = 1$ implies that $\mathbf{x}$ is considered as 'adversarial', i.e., perturbed, and if s is a similarity score, the opposite decision is made.

A detection method can act on the model to be protected by modifying its training procedure using tools such as reverse cross-entropy [PANG and collab., 2018] or the rejection option [ALDAHDOOH and collab., 2021b; SOTGIU and collab., 2020]. In that case, both detector and model are trained jointly. Those methods are usually vulnerable to changes in attack mechanisms, and thus, they need global re-training if a modification of the detector is introduced. On the other hand, it is also possible to craft detectors on top of a fixed trained model. Those methods can be divided into two main categories: supervised methods, where the detector knows the attack that

will be perpetrated, and unsupervised methods, where the detector can only rely on clean samples, which is not desired in practice.

Generally, supervised methods use simple machine learning algorithms (e.g., SVM or a logistic regressor) to distinguish between natural and adversarial examples. The effectiveness of such methods heavily relies on natural and adversarial feature extraction. They can be extracted directly from the network's layers [Carrara and collab., 2018; Lu and collab., 2017; Metzen and collab., 2017], or using statistical tools (e.g., maximum mean discrepancy [Grosse and collab., 2017], PCA [Li and Li, 2017], kernel density estimation [Feinman and collab., 2017], local intrinsic dimensionality [Ma and collab., 2018], model uncertainty [Feinman and collab., 2017] or natural scene statistics [Kherchouche and collab., 2020]). Supervised methods, which heavily depend on the knowledge about the perpetrated attack, tend to overfit to that attack mechanism and usually generalize poorly.

Unsupervised methods do not assume any knowledge of the attacker. Indeed, new attack mechanisms are crafted every year, and it is unrealistic to assume knowledge about the attacker. To overcome that absence of prior knowledge about the attacker, unsupervised methods can only rely on natural samples. The features extraction rely on different techniques, such feature squeezing [Xu and collab., 2018], adaptive noise, [Liang and collab., 2021], using denoising autoencoders [Meng and Chen, 2017], network invariant [Ma and collab., 2019] or training an auxiliary model [Aldahdooh and collab., 2021b; Sotgiu and collab., 2020; Zheng and Hong, 2018]. Raghuram and collab. [2021] uses dimension reduction, kNN and layer aggregation to distinguish between natural and adversarial samples. In this paper, we only focus on unsupervised methods that cannot act on the model to be protected.

7.3 Our Proposed Detector

7.3.1 Background on data depth

The notion of data depth goes back to John Tukey in 1975, who introduced the halfspace depth [Tukey, 1975]. Data depth functions are useful nonparametric statistics allowing to rank elements of a multivariate space $\mathbb{R}^d$ w.r.t. a probability distribution (or a dataset). Given a random variable $\mathbf{Z}$ which follows the distribution $p_{\mathbf{Z}}$, a data depth can be defined as:

$$\begin{aligned} \mathrm{D}: \quad \mathbb{R}^d \times \mathscr{P}(\mathbb{R}^d) &\longrightarrow [0,1]\,, \\ (\mathbf{z}, p_{\mathbf{Z}}) &\longmapsto \mathrm{D}(\mathbf{z}, p_{\mathbf{Z}}). \end{aligned} \tag{7.4}$$

The higher the value of the depth function, the deeper the element is in the reference distribution. Various data depths have been introduced over the year

(see Chapter 2 of STAERMAN [2022] for a survey), including halfspace depth [TUKEY, 1975], the simplicial depth [LIU, 1990], the projection depth [LIU, 1992] or the zonoid depth [KOSHEVOY and MOSLER, 1997]. Despite their applications in statistics and machine learning (e.g., regression [HALLIN and collab., 2010; ROUSSEEUW and HUBERT, 1999], classification [MOZHAROVSKYI and collab., 2015], automatic text evaluation [STAERMAN and collab., 2021b] or anomaly detection [ROUSSEEUW and HUBERT, 2018; SERFLING, 2006; STAERMAN and collab., 2022, 2020]) the use of data depth with representation models, and more generally to deep learning, remains overlooked by the community. The halfspace depth is the first and the most studied depth in the literature probably due to its appealing properties [DONOHO and GASKO, 1992; ZUO and SERFLING, 2000] as well as its connections with univariate quantiles. However, it suffers from computational burden in practice [DYCKERHOFF and MOZHAROVSKYI, 2016; ROUSSEEUW and STRUYF, 1998]. Indeed, it requires solving an optimization problem over the unit hypersphere of a non-differentiable quantity. To remedy this drawback, [RAMSAY and collab., 2019] introduced the Integrated Rank-Weighted (IRW) depth (see also CHEN and collab. [2015]; STAERMAN and collab. [2021a]), which involves an expectation as an alternative to the infimum over the unit hypersphere of the halfspace depth, making it easier to compute. The IRW depth is scale and translation invariant and has been successfully applied to anomaly detection [CHEN and collab., 2015; STAERMAN and collab., 2021a] making it a good candidate for our purposes. Formally, the IRW depth is defined as:

$$\mathrm{D}_{\mathrm{IRW}}(\mathbf{z}, p_{\mathbf{Z}}) = \int_{\mathbb{S}^{d-1}} \min\left\{\mathrm{F}_{\mathbf{u}}\left(\langle \mathbf{u}, \mathbf{z}\rangle\right), 1 - \mathrm{F}_{\mathbf{u}}\left(\langle \mathbf{u}, \mathbf{z}\rangle\right)\right\} \mathrm{d}\mathbf{u}, \tag{7.5}$$

where the unit hypersphere is denoted as $\mathbb{S}^{d-1}$ and $\mathrm{F}_{\mathbf{u}}(t) = \mathbb{P}(\langle \mathbf{u}, \mathbf{Z}\rangle \leq t)$. A Monte-Carlo scheme is used to approximate the integral by an empirical mean. Given a training dataset $\mathscr{S}_n = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ following $p_{\mathbf{Z}}$ and $\mathbf{u}_k \in \mathbb{S}^{d-1} \forall k \in \{1, \cdots, n_{\mathrm{proj}}\}$,the empirical version of the IRW depth, which can be computed in $\mathscr{O}(n_{\mathrm{proj}} n d)$ and is then linear in all of its parameters, is defined as:

$$\widetilde{\mathrm{D}}^{\mathrm{MC}}_{\mathrm{IRW}}(\mathbf{z}, \mathscr{S}_n) = \frac{1}{n_{\mathrm{proj}}} \sum_{k=1}^{n_{\mathrm{proj}}} \min\left\{\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{\langle \mathbf{u}_k, \mathbf{z}_i - \mathbf{z}\rangle \leq 0\}, \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{\langle \mathbf{u}_k, \mathbf{z}_i - \mathbf{z}\rangle > 0\}\right\}, \tag{7.6}$$

7.3.2 APPROVED: Our depth-based detector

Intuition. Our detector tries to answer this simple question: can we find a metric that will be able to distinguish between natural and arbitrary adversarial samples? At the logit layer, we want to compare the new input to the training samples of its predicted class to measure whether the new sample is behaving as expected. Data depths, particularly the IRW depth, are serious candidates as they measure the 'distance'

between a given new input to the training probability distribution.

APPROVED **in a nutshell.** To detect whether a given model $f_{\boldsymbol{\theta}}$ can trust a new input $\mathbf{x}$, APPROVED will perform three steps:

1. **Logits computation**. For an new input $\mathbf{x}$, APPROVED first require to extract the logits (i.e., $f_{\boldsymbol{\theta}}^{L-1}(\mathbf{x})$) from the pretrained classifier.

2. **Similarity score computation**. APPROVED relies on the IRW depth score $D_{\text{IRW}}(f_{\boldsymbol{\theta}}^{L-1}(\mathbf{x}), p_{\hat{y}}^{L-1})$, between $p_{\hat{y}}^{L-1}$, the training distribution of the predicted class $\hat{y} = f_{\boldsymbol{\theta}}^{L}(\mathbf{x})$ at the logit layer, and $f_{\boldsymbol{\theta}}^{L-1}(\mathbf{x})$, using Algorithm 8 in Section C.2 to evaluate (7.6).

3. **Thresholding.** For a given threshold γ, the test input sample $\mathbf{x}$ is detected as clean if $D_{\text{IRW}}(f_{\boldsymbol{\theta}}^{L-1}(\mathbf{x}), p_{\hat{y}}^{L-1}) > \gamma$, otherwise, it is considered as adversarial. A classical way to select γ it by selecting an amount of training samples the detector can wrongfully detect.

7.3.3 Comparison with existing detectors

We benchmark our approach with two unsupervised detection methods: FS and MagNet. We chose these baselines because they are unsupervised and do not modify the model to protect. We could consider NIC [Ma and Liu, 2019] but extracting features at each layer is computationally expensive.

The Feature Squeezing method (FS; [Xu and collab., 2018]). It computes the feature squeezing of the input, extracts its prediction, and compares it to the original prediction. The further away they are, the more likely the input is adversarial. In practice, four versions of the input are needed: the original input, a low-precision version, a median-filtered version, and a denoising-filtered version. One inference on the model is required for each of the four inputs. Later, the maximal L_1 difference between the original prediction and each of the other three is picked. FS is, therefore, parameter-free and does not require training. However, the necessary time to extract the essential features and the memory needed to store all the input modifications and their prediction are quite high.

MagNet [Meng and Chen, 2017]. It is based on the training of two components: first, a detector that detects if a sample $\mathbf{x}$ is clean or not, then, a reformer that tries to find an approximation of the input closer to the training manifold. In practice, each of those components is an autoencoder that must be trained on clean samples before testing new inputs. MagNet requires three inferences at testing time, one on the detector, one on the reformer, and one on the original model to protect. Therefore, even though, at inference time, little time is necessary to output the prediction, MagNet requires careful training, which is time-consuming, and storing two autoencoders, which is highly memory-consuming.

APPROVED, similarly to FS and contrary to MagNet, does not require training time and is parameter-free. Contrary to FS, it only requires one inference on the model to extract the logits of the input. It is, therefore, *less computationally and time-consuming*. The summary of computational time and resources needed to deploy each detection method is provided in Section C.3. Finally, since data depths are non-differentiable, it is not straightforward for gradient-based attacks that have full access to the detection method to attack APPROVED.

7.4 Adversarial Attacks on Vision Transformers (ViT)

In the following, we provide insights on the behavior of vision transformers under the threat of adversarial attacks, along with a comparison to the classically used ResNets models.

7.4.1 Set-up

Datasets and classifiers. We conducted our study on pre-trained Vision Transformers. We rely on three widely used vision datasets: CIFAR10 [KRIZHEVSKY, 2009], CIFAR100 and TinyImageNet (Tiny) JIAO and collab. [2019]. Training details can be found in Section C.1.

Model	Dataset	Acc (%)
ViT	CIFAR10	98.7
	CIFAR100	92.4
	Tiny ImageNet	86.4

Table 7.1: ViT accuracy for each dataset

Performance measures. We use two different metrics to compare the different detection methods:

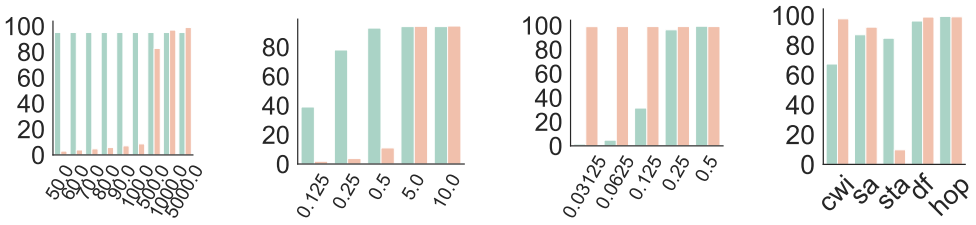
AUROC↑: Area Under the Receiver Operating Characteristic curve [DAVIS and GOADRICH, 2006]. It represents the relation between True Positive Rates (TPR), i.e., the percentage of perturbed samples detected as adversarial, and False Positive Rates (FPR), i.e., the percentage of clean samples detected as adversarial. The higher the AUROC↑ is, the better the detector's performances are.

$FPR{\downarrow}_{90\%}$*:* False Positive Rate at 90% True Positive Rate. It represents the number of natural samples detected as adversarial when 90% of the attacked samples are detected. Lower is better.

Remark. We discard the perturbed samples that do not fool the underlying classifier. Indeed, detecting a sample that does not perturb the classifier's functioning as either natural or adversarial is a valid answer.

Attacks. For the experiments, we will evaluate the different detection methods on the attacks presented in Subsection 7.2.1. Under L_1-norm constraint, we craft attacks following PGD^1 scheme. For the L_2-norm constraint, we consider PGD^2, DF and HOP. Under L_∞-norm constraint, we study PGD^∞, BIM and FGSM attacks, CW^∞ and SA. Finally, we create STA attacks, which are not subject to a norm constraint. The values of the maximal allowed perturbation are discussed in the next section.

7.4.2 Adversarial attack calibration



(a) PGD_1 (b) PGD_2 (c) PGD_∞ (d) others

Figure 7.1: Percentage of successful attacks depending on the L_p-norm constraint, the maximal perturbation ε and the attack algorithm on ResNet18 (green) and ViT (orange).

Given that the variety of attacks comes from choosing the L_p-norm constraint and the maximal allowed L_p-norm perturbation ε, it is crucial to select them carefully. Adversarial attacks and defense mechanisms have been widely studied for classical convolutional networks, particularly for ResNets [GOODFELLOW and collab., 2014; MADRY and collab., 2018; MENG and CHEN, 2017; MOOSAVI-DEZFOOLI and collab., 2016; XU and collab., 2018; ZHANG and collab., 2019]. Hence, choosing the maximally allowed perturbation ε for ViT comes naturally from comparing the success attack rates between attacks on ResNets and ViTs.

In Figure 7.1, we present the success rates for attacks on Resnet18 (resp. on ViT) in blue (resp. in orange), for different attack mechanisms, different L_p-norms and different maximal perturbation ε (the results for FGSM and BIM are relegated to Section C.4). Attacks behave differently on ResNets and on ViTs: on L_∞-norm constraints, at equal ε, the attacks are more potent on the ViT than on ResNet18. Indeed, the input of a ViT has more pixels than the input of a ResNet ($32 \times 32 \times 3$ for ResNet and $224 \times 224 \times 3$ for ViT). Limiting the perturbation by an L_∞-norm constraint, i.e., controlling the maximal perturbation pixel-wise without controlling the number of modified pixels, will create samples further away from the original sample if it has more pixels. On the contrary, under L_1 and L_2-norm constraints, the opposite behavior is observable: at fixed ε, the attack are more potent on ResNets than on ViTs. This can be explained by the fact that limiting L_1 or L_2-norm perturbations controls the average perturbations on the whole input sample. The modifications are therefore smaller pixel-wise if the image is bigger. While on ResNets, the classical values of ε are lower than 40 on L_1-norm constraints and 2 on L_2-norm-constraints, we had to increase the maximum ε studied for those L_p-norm constraint to have successful enough attacks. Finally, Spatial Transformation Attacks (STA) disturb ResNets' functioning more easily than ViTs'.

Summary. To sum up, in the remaining of the paper, under L_1-norm constraint, we craft PGD^1 attacks with maximum norm constraint $\varepsilon \in \{50, 60, 70, 80, 90, 100, 500, 1000, 5000\}$. For the L_2-norm, we consider PGD^2 with $\varepsilon \in \{0.125, 0.25, 0.5, 5, 10\}$, DF

with no ε, and HOP attacks with 3 restarts and ε = 0.1. Under L_∞-norm constraint, we consider PGD$^\infty$, BIM and FGSM attacks with $\varepsilon \in \{0.03125, 0.0625, 0.125, 0.25, 0.5\}$, CW$^\infty$ with ε = 0.3125 and SA with ε = 0.125. Finally, STA attacks, which are not subject to a norm constraint, can rotate the input up to 60°, and translate it up to 10 pixels.

7.4.3 Locating the relevant information

In the previous section, we saw that the attacks behave differently w.r.t. the classifier on which they are perpetrated. We now continue this investigation by looking at the differences between the two models from the depth scores' perspective. In this framework, we define the layer to have relevant information when the difference between the depth score on the naturals and the depth score on the adversarial is significant. Indeed, the higher the difference, the more evident the shift between the distributions of the natural and the adversarial induced by the depth score will be, and hence the easier it will be to find a threshold that distinguishes natural from adversarial samples. We start by computing layer per layer the differences between the IRW depth on the natural samples ($D_{IRW}(f_{\boldsymbol{\theta}}^{\ell}(\mathbf{x}); p_{\hat{y}}^{\ell})$) and on the adversarial samples ($D_{IRW}(f_{\boldsymbol{\theta}}^{\ell}(\mathbf{x}'); p_{\hat{y}}^{\ell})$) both for ViT and for ResNet18. In Figure 7.2, we plot the mean and standard deviation for each layer and each network.The diamond points represent the outliers. Figure 7.2 shows that the information about whether a sample is natural or adversarial, based on the study of the IRW depth, is significantly spread across the ResNet18 model: in each layer, the values range between 0 and 0.06. On the contrary, on ViT, this information is concentrated in the logit layer, where the values range between 0.05 and 0.2 while the values range from 0 to 0.05 for the other layers. To summarize, while relevant information to distinguish between natural and adversary samples is diffused in the ResNet18 model, which has small and similar values for all the layers, the most valuable information is instead concentrated at the logit layer for the ViT network, which experiences larger values only for that particular layer. It seems, therefore,

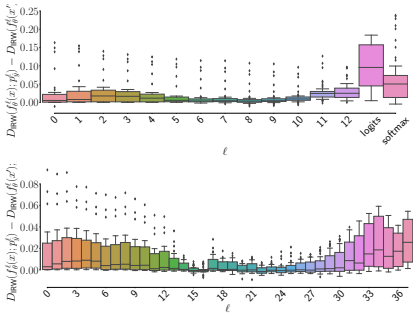


Figure 7.2: Difference between natural and adversarial IRW depth values as a function of the layer on ViT (top) and on ResNet18 (bottom), averaged over the attacks.

Table 7.2: Averaged results over the different attacks for each considered L_p-Norm constraints for APPROVED, FS and MagNet, along with the Averaged results over the norms. The results are presented as mean $\pm standard\_deviation$. The best results are presented in **bold**.

	APPROVED						FS						MagNet					
	CIFAR10		CIFAR100		Tiny		CIFAR10		CIFAR100		Tiny		CIFAR10		CIFAR100		Tiny	
	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$
L_1	**94.0** ±0.2	**13.2** ±3.5	**78.3** ±7.6	**46.4** ±13.1	**75.2** ±1.3	**59.2** ±2.7	79.5 ±3.3	34.9 ±8.3	71.1 ±5.1	55.5 ±8.0	54.2 ±14.0	75.1 ±11.0	51.3 ±1.1	91.1 ±0.9	50.1 ±0.2	90.2 ±0.2	49.4 ±0.3	90.0 ±1.4
L_2	**94.1** ±0.7	**14.6** ±3.5	**80.5** ±4.9	**44.0** ±11.3	**76.8** ±4.6	**53.8** ±10.1	77.3 ±1.8	37.2 ±8.6	68.2 ±5.1	58.9 ±10.3	61.8 ±12.0	72.4 ±10.6	51.0 ±1.2	89.8 ±2.7	50.6 ±0.7	89.3 ±2.0	49.9 ±1.4	89.0 ±2.5
L_∞	**95.3** ±0.5	**13.4** ±10.9	**86.7** ±0.4	**29.9** ±19.3	**91.8** ±0.8	**19.1** ±19.6	73.0 ±3.3	53.4 ±18.3	62.6 ±6.8	67.3 ±11.6	74.6 ±17.8	61.2 ±25.9	56.2 ±9.7	80.0 ±7.4	55.0 ±0.7	81.3 ±15.8	50.9 ±2.6	88.3 ±4.3
no Norm	**94.9** ±0.0	**10.5** ±0.0	**87.4** ±0.0	**32.1** ±0.0	**80.2** ±0.0	**42.5** ±0.0	78.8 ±0.0	37.5 ±0.0	65.4 ±0.0	50.0 ±0.0	53.0 ±0.0	77.5 ±0.0	39.4 ±0.0	93.5 ±0.0	38.3 ±0.0	92.8 ±0.0	34.9 ±0.0	95.6 ±0.0
Average	**94.7** ±5.6	**13.5** ±7.5	**83.2** ±8.9	**37.2** ±7.8	**83.9** ±10.3	**37.7** ±23.8	75.8 ±4.2	44.2 ±16.4	66.1 ±7.0	62.0 ±11.8	65.8 ±18.0	67.7 ±19.6	53.3 ±7.7	85.3 ±13.5	52.3 ±7.0	85.7 ±12.0	49.8 ±3.4	89.2 ±3.6

Table 7.3: Averaged results over the different types of attack mechanism for APPROVED, FS, and MagNet, along with the averaged results over the norms. The results are presented as mean $_{\pm standard\_deviation}$. The best results are presented in **bold**. Dashed values (–) corresponds to attacks that take more than 48 hours to run on V100 GPUs.

	APPROVED						FS						MagNet					
	CIFAR10		CIFAR100		Tiny		CIFAR10		CIFAR100		Tiny		CIFAR10		CIFAR100		Tiny	
	AUROC↑	FPR$\downarrow_{90\%}$	AUROC↑	FPR$\downarrow_{90\%}$	AUROC↑	FPR$\downarrow_{90\%}$	AUROC↑	FPR$\downarrow_{90\%}$	AUROC↑	FPR$\downarrow_{90\%}$	AUROC↑	FPR$\downarrow_{90\%}$	AUROC↑	FPR$\downarrow_{90\%}$	AUROC↑	FPR$\downarrow_{90\%}$	AUROC↑	FPR$\downarrow_{90\%}$
PGD	**95.5** ±4.3	**9.6** ±10.8	**81.3** ±7.8	**41.2** ±14.6	**81.0** ±10.2	**45.0** ±26.1	77.2 ±3.8	44.4 ±14.2	70.1 ±4.9	62.5 ±11.1	65.6 ±19.0	66.2 ±22.5	51.5 ±1.4	89.7 ±2.7	62.4 ±13.3	68.1 ±24.1	49.5 ±0.7	90.0 ±1.1
BIM	**96.8** ±4.4	**7.1** ±10.1	**82.1** ±13.1	**37.9** ±26.2	**95.0** ±7.4	**11.8** ±16.6	71.2 ±1.5	69.6 ±2.9	64.3 ±1.9	77.8 ±1.4	86.5 ±2.2	60.4 ±19.2	52.6 ±2.2	86.0 ±3.5	51.8 ±1.6	86.9 ±2.7	49.9 ±0.2	89.7 ±0.3
FGSM	**90.5** ±8.8	**29.7** ±29.4	**90.4** ±8.4	**23.9** ±15.8	**85.6** ±7.2	**33.5** ±14.6	73.7 ±8.3	32.7 ±5.0	54.8 ±6.2	56.0 ±5.4	52.8 ±5.3	75.1 ±2.3	62.6 ±11.1	65.2 ±28.4	62.4 ±13.3	68.1 ±24.1	52.9 ±4.9	85.4 ±0.8
HOP	**98.3** ±0.0	**3.3** ±0.0	**89.1** ±0.0	**24.8** ±0.0	**87.1** ±0.0	**31.8** ±0.0	74.5 ±0.0	25.0 ±0.0	62.7 ±0.0	50.0 ±0.0	59.1 ±0.0	76.3 ±0.0	53.4 ±0.0	83.6 ±0.0	50.0 ±0.0	89.9 ±0.0	52.7 ±0.0	83.8 ±0.0
DeepFool	**86.5** ±0.0	45.4 ±0.0	**75.5** ±0.0	59.9 ±0.0	–	–	79.7 ±0.0	**31.2** ±0.0	62.2 ±0.0	**50.0** ±0.0	–	–	50.3 ±0.0	89.7 ±0.0	50.0 ±0.0	89.9 ±0.0	–	–
SA	**98.2** ±0.0	**3.3** ±0.0	**89.6** ±0.0	**26.0** ±0.0	**77.0** ±0.0	**49.1** ±0.0	72.0 ±0.0	25.0 ±0.0	63.3 ±0.0	50.0 ±0.0	48.7 ±0.0	78.5 ±0.0	55.1 ±0.0	82.4 ±0.0	54.9 ±0.0	82.6 ±0.0	50.6 ±0.0	89.4 ±0.0
CW	**90.4** ±0.0	**30.6** ±0.0	**81.7** ±0.0	**42.2** ±0.0	–	–	78.8 ±0.0	37.5 ±0.0	67.0 ±0.0	50.0 ±0.0	–	–	50.6 ±0.0	89.3 ±0.0	50.0 ±0.0	89.8 ±0.0	–	–
STA	**94.9** ±0.0	**10.5** ±0.0	**87.4** ±0.0	**32.1** ±0.0	**80.2** ±0.0	**42.5** ±0.0	78.8 ±0.0	37.5 ±0.0	65.4 ±0.0	50.0 ±0.0	53.0 ±0.0	77.5 ±0.0	39.4 ±0.0	93.5 ±0.0	38.3 ±0.0	92.8 ±0.0	34.9 ±0.0	95.6 ±0.0

relevant to build a detector *specific* for vision transformers based *only* on the output of the logit layer.

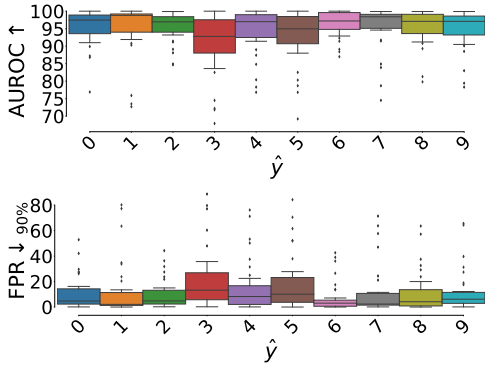


Figure 7.3: APPROVED's AUROC↑ and FPR$\downarrow_{90\%}$ per class, averaged over CIFAR10.

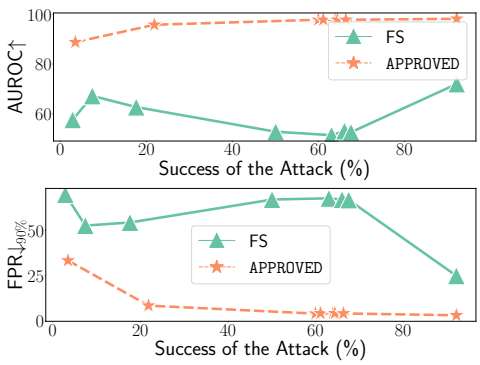


Figure 7.4: Detector Performances under blackbox Adaptive Attack.

7.5 Experiments

7.5.1 Results

Performances of APPROVED compared to other unsupervised detection methods. In Table C.3, Table C.4, and Table C.5 relegated to Section C.5 , we report the detailed results for each considered detection method under the threat of each attack mechanism, L_p-norm constraint and maximum perturbation ε. In Table 7.2, we report the averaged AUROC↑ and FPR$\downarrow_{90\%}$ on each of the considered L_p-norm, along with the global average for each detector, on CIFAR10, CIFAR100, and Tiny ImageNet. Overall, APPROVED shows better results than the SOTA detection methods. On CIFAR10, APPROVED creates an increase of AUROC↑ of 18.9% and a decrease of FPR$\downarrow_{90\%}$ of 30.7% compared to the best performing state-of-the-art detector, i.e., FS. On CIFAR100, the improvements are 17.1% and 24.8%, respectively, while they are 18.0% and 29.9% on Tiny ImageNet. In addition, FS and APPROVED have similar dispersions. Moreover, under specific L_p-norm constraints, our method consistently outperforms SOTA methods, especially under the L_∞-norm constraint where APPROVED outperforms FS (resp. MagNet) by 22.3% (resp. 39.1%) in terms of AUROC↑ and 40.0% (resp. 66.6%)

in terms of FPR$\downarrow_{90\%}$ on CIFAR10. Finally, by looking at the detailed results presented in Table C.3 and Table C.4, we can deduce that FS and APPROVED have opposite behaviors: when the performances of FS decrease, APPROVED's performances tend to improve. For example, under the L_∞-norm constraint, APPROVED has more trouble detecting attacks with small perturbations, while FS has more difficulty detecting attacks with large perturbations. Indeed, since APPROVED measures the depth of a sample within a distribution, it will be able to recognize the strongest attacks well.

Performances per attack. In Table 7.3 we give the overall idea of the results on all three datasets per attack mechanism by showing them in terms of *mean* and *standard deviation* (std) on the AUROC$\uparrow$ and on the FPR$\downarrow_{90\%}$. APPROVED turns out to consistently outperform the state-of-the-art detectors forall datasets. In particular, we notice that the FGSM attacks that are the easiest to generate are the ones that present the highest diversity among the results in the methods examined. Indeed, by looking at Table 7.3, we can find larger values of the standard deviation in correspondence to that attack. Moreover, APPROVED is capable of recognizing attacks that are more difficult for the competitors (e.g., STA for MagNet or FGSM for FS). We also observe that for APPROVED the most challenging task is to distinguish natural and adversarial samples when they are crafted with DeepFool. However, it is the best choice even in this case as it reaches better performances than the other detectors.

7.5.2 Adaptive attacks

In this experiment, we evaluate APPROVED against adaptive attacks, which has knowledge about the defense [ATHALYE and collab., 2018; CARLINI and WAGNER, 2017; TRAMER and collab., 2020]. Two scenarios can be considered with adaptive attacks: whitebox and blackbox. Whitebox attacks (*e.g.* BPDA [ATHALYE and collab., 2018]) are not straightforward to adapt in our case since finding a differentiable surrogate of IRW remains a very challenging open research question in the statistical community, which has never been tackled. As a matter of fact, the only attempts to approximate a non-differentiable depth was performed not on the IRW depth but on the Tuckey depth in DYCKERHOFF and collab. [2021], with very poor results as pointed out in SHE and collab. [2021]. Thus, in this experiment, we rely on blackbox attacks and present the results in Figure 7.4. We attacked both APPROVED and FS using a modified version of SA [ENGSTROM and collab., 2019], for which the attack objective has been modified to allow the attacker to fool both the detection method as well as the classifier. We rely on an hyperparameter α that weights the relative importance of the two parts of the objective. It is straightforward (cf. Figure 7.4) that APPROVED is less sensitive to adaptive attacks than FS. This results further validates the use of the IRW depth to craft detection method and further assesses the superiority of APPROVED.

7.5.3 Finer analysis

Per class analysis. As explained in Subsection 7.3.2, APPROVED is based on the IRW depth, which computes the depth score of the sample w.r.t. the original distribution by class. Figure 7.3 shows the per-class performances averaged over the different attacks on CIFAR10, while Figure C.2, relegated to Section C.6 due to space constraints, shows the performances on CIFAR100. It is clear from Figure 7.3 that APPROVED does not have equal performances on every class. In particular, some classes present extremely high mean average AUROC↑ (i.e., class 7), others exhibit very low FPR$\downarrow_{90\%}$ (i.e., class 6), while some others have their adversarial and clean samples tough to distinguish (i.e., class 3 and 5). The same behavior is observable of CIFAR100 (see Figure C.2).

AUROC↑ vs FPR$\downarrow_{90\%}$. We conclude our analysis by looking at the trade-off between AUROC↑and FPR$\downarrow_{90\%}$(see Figure 7.5). The ideal method would concentrate the results on the upper left of the figure, which corresponds to high AUROC↑ and low FPR$\downarrow_{90\%}$, while a poor detector would concentrate them in the bottom right corner of the figure, which corresponds to low AUROC↑ and high FPR$\downarrow_{90\%}$.

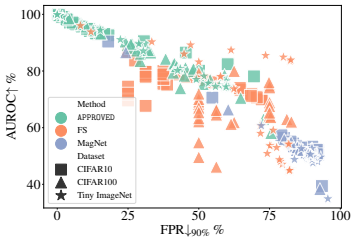


Figure 7.5: AUROC↑ as a function of FPR$\downarrow_{90\%}$ for APPROVED, FS, and MagNet on all considered datasets.

We observe that on CIFAR10, the APPROVED points are more concentrated in the upper left corner, while the FS points are concentrated in the center of the figure and MagNet's in the lower right corner. On CIFAR100 and Tiny ImageNet, the results for our method are slightly more spread out in the top left and center of the figure, while for FS and MagNet, they are still in the center and bottom right, respectively. Note that FS has a different behavior than expected, i.e., the line connecting the top left corner with the bottom right corner. This behavior change can be observed for FPR$\downarrow_{90\%}$ between 25%-35% on CIFAR10 and between 50%-75% on CIFAR100 and Tiny ImageNet. On CIFAR10, FS presents a lower AUROC↑ for a fixed FPR$\downarrow_{90\%}$ than expected, whereas, on CIFAR100, it presents a lower AUROC↑ (for FPR$\downarrow_{90\%}$ values between 50%-60%) or higher (for FPR$\downarrow_{90\%}$ values around 75%) than expected.

7.6 Conclusions and Limitations

We introduced APPROVED, an efficient unsupervised detection method designed to defend against adversarial attacks. In contrast with previous detection methods, which were built for ResNet architectures, APPROVED is well suited for vision transformers which nowadays represent the state-of-the-art. While the relevant information about the discrepancy between clean and adversarial samples is distributed across all layers

of ResNets, for the transformers, it was empirically shown to be concentrated in the logit layer. This motivated us to build APPROVED on top of this logit layer by computing a similarity score between an input sample and the training distribution based on the statistical notion of *data depth*. We chose to use the Integrated Rank-Weighted depth, which lends itself to fast inference computations and is non-differentiable, making it harder for gradient-based adversarial methods to craft malicious samples. We conduct extensive numerical experiments and prove that APPROVED outperforms the other state-of-the-art methods significantly.

Future Research. We think our method paves the way for future research efforts. Indeed, there is still room for improvement: even if the AUROC↑ performances are good, the $\text{FPR}\downarrow_{90\%}$ are also fairly high. We believe the idea of leveraging information contained in layers of transformers through data depths can be fruitful in improving defense mechanisms against adversarial attacks. Our research is expected to have a positive societal impact by protecting the integrity of AI systems, especially necessary in critical systems such as autonomous cars [MORGULIS and collab., 2019] or stock predictions [XIE and collab., 2022].

Chapter 7 Conclusion

In this work, we presented an efficient unsupervised detection method designed to defend Vision Transformers from adversarial attacks. While distributed across all layers of ResNets, the information about the difference between natural and attacked samples is concentrated in the logit layer for the transformers. This motivated us to build a detection method on top of this layer by computing a similarity score between an input sample and the training distribution based on the statistical notion of data depth. We chose to use the Integrated Rank-Weighted depth, which lends itself to fast inference computations and is non-differentiable, making it harder for gradient-based adversarial methods to craft malicious samples. Experimentally, we proved that APPROVED outperforms the other state-of-the-art methods significantly.

7.7 References

ALAYRAC, J.-B., J. UESATO, P.-S. HUANG, A. FAWZI, R. STANFORTH and P. KOHLI. 2019, «Are labels required for improving adversarial robustness?», in *Advances in Neural Information Processing Systems*, p. 12 214–12 223. 196

ALDAHDOOH, A., W. HAMIDOUCHE and O. DEFORGES. 2021a, «Reveal of vision transformers robustness against adversarial attacks», *arXiv preprint arXiv:2106.03734*. 196

ALDAHDOOH, A., W. HAMIDOUCHE and O. DÉFORGES. 2021b, «Revisiting model's uncertainty and confidences for adversarial example detection», *arXiv preprint arXiv: 2103.05354*. 196, 197

ALDAHDOOH, A., W. HAMIDOUCHE, S. A. FEZZA and O. DÉFORGES. 2021c, «Adversarial example detection for DNN models: A review», *arXiv preprint arXiv:2105.00203*. 193

ALI, M., Y.-F. HU, D. K. LUONG, G. OGUNTALA, J.-P. LI and K. ABDO. 2020, «Adversarial attacks on ai based intrusion detection system for heterogeneous wireless communications networks», in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, IEEE, p. 1–6. 193

ALVES, E., D. BHATT, B. HALL, K. DRISCOLL, A. MURUGESAN and J. RUSHBY. 2018, «Considerations in assuring safety of increasingly autonomous systems», *NASA*. 192

AMODEI, D., C. OLAH, J. STEINHARDT, P. CHRISTIANO, J. SCHULMAN and D. MANÉ. 2016, «Concrete problems in ai safety», *arXiv preprint arXiv:1606.06565*. 193

ANDRIUSHCHENKO, M., F. CROCE, N. FLAMMARION and M. HEIN. 2020, «Square attack: a query-efficient black-box adversarial attack via random search», in *European Conference on Computer Vision*, Springer, p. 484–501. 196

ATHALYE, A., N. CARLINI and D. WAGNER. 2018, «Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples», in *International conference on machine learning*, PMLR, p. 274–283. 204

ATZMON, M., N. HAIM, L. YARIV, O. ISRAELOV, H. MARON and Y. LIPMAN. 2019, «Controlling neural level sets», in *Advances in Neural Information Processing Systems*, p. 2034–2043. 196

BENZ, P., S. HAM, C. ZHANG, A. KARJAUV and I. S. KWEON. 2021, «Adversarial robustness comparison of vision transformer and mlp-mixer to cnns», *arXiv preprint arXiv:2110.02797*. 196

BOJARSKI, M., D. DEL TESTA, D. DWORAKOWSKI, B. FIRNER, B. FLEPP, P. GOYAL, L. D. JACKEL, M. MONFORT, U. MULLER, J. ZHANG and collab.. 2016, «End to end learning for self-driving cars», *arXiv preprint arXiv:1604.07316*. 193

BREUNIG, M., H.-P. KRIEGEL, R. NG and J. SANDER. 2000, «Lof: Identifying density-based local outliers», in *ACM SIGMOD*, vol. 29, ACM, p. 93–104. 193

CARLINI, N. and D. WAGNER. 2017, «Towards evaluating the robustness of neural networks», in *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, p. 39–57. 195, 204

CARMON, Y., A. RAGHUNATHAN, L. SCHMIDT, J. C. DUCHI and P. S. LIANG. 2019, «Unlabeled data improves adversarial robustness», in *Advances in Neural Information Processing Systems*, p. 11 192–11 203. 196

CARRARA, F., R. BECARELLI, R. CALDELLI, F. FALCHI and G. AMATO. 2018, «Adversarial examples detection in features distance spaces», in *Computer Vision - ECCV 2018 Workshops - Munich, Germany, Proceedings, Part II*, vol. 11130, Springer, p. 313–327. 197

CHANDOLA, V., A. BANERJEE and V. KUMAR. 2009, «Anomaly detection: A survey», *ACM Comput. Surv.*, vol. 41, nº 3, p. 15:1–15:58, ISSN 0360-0300. 193

CHEN, B., K. M. TING, T. WASHIO and G. HAFFARI. 2015, «Half-space mass: a maximally robust and efficient data depth method», *Machine Learning*, vol. 100, nº 2, p. 677–699. 198

CHEN, J., M. I. JORDAN and M. J. WAINWRIGHT. 2020, «Hopskipjumpattack: A query-efficient decision-based attack», in *2020 ieee symposium on security and privacy (sp)*, IEEE, p. 1277–1294. 196

CHEN, X., C.-J. HSIEH and B. GONG. 2021, «When vision transformers outperform resnets without pretraining or strong data augmentations», *arXiv preprint arXiv:2106.01548*. 194

CHOW, C.-K. 1957, «An optimum character recognition system using decision functions», *IRE Transactions on Electronic Computers*, , nº 4, p. 247–254. 193

CROCE, F. and M. HEIN. 2020, «Minimally distorted adversarial examples with a fast adaptive boundary attack», in *International Conference on Machine Learning*, PMLR, p. 2196–2205. 193

DAVIS, J. and M. GOADRICH. 2006, «The relationship between precision-recall and roc curves», in *Proceedings of the 23rd international conference on Machine learning*, p. 233–240. 200

DENG, Y. and L. J. KARAM. 2020a, «Frequency-tuned universal adversarial attacks», *arXiv preprint arXiv:2003.05549*. 193

DENG, Y. and L. J. KARAM. 2020b, «Towards imperceptible universal attacks on texture recognition», *arXiv preprint arXiv:2011.11957*. 193

DONG, Y., T. PANG, H. SU and J. ZHU. 2019, «Evading defenses to transferable adversarial examples by translation-invariant attacks», in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 4312–4321. 193

DONOHO, D. L. and M. GASKO. 1992, «Breakdown properties of location estimates based on half space depth and projected outlyingness», *The Annals of Statistics*, vol. 20, p. 1803–1827. 198

DOSOVITSKIY, A., L. BEYER, A. KOLESNIKOV, D. WEISSENBORN, X. ZHAI, T. UNTERTHINER, M. DEHGHANI, M. MINDERER, G. HEIGOLD, S. GELLY, J. USZKOREIT and N. HOULSBY. 2021, «An image is worth 16x16 words: Transformers for image recognition at scale», *ICLR*. 194

DUAN, R., X. MA, Y. WANG, J. BAILEY, A. K. QIN and Y. YANG. 2020, «Adversarial camouflage: Hiding physical-world attacks with natural styles», in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, p. 1000–1008. 193

DYCKERHOFF, R. and P. MOZHAROVSKYI. 2016, «Exact computation of the halfspace depth», *Computational Statistics & Data Analysis*, vol. 98, p. 19–30. 198

DYCKERHOFF, R., P. MOZHAROVSKYI and S. NAGY. 2021, «Approximate computation of projection depths», *Computational Statistics & Data Analysis*, vol. 157, p. 107 166. 204

ENGSTROM, L., B. TRAN, D. TSIPRAS, L. SCHMIDT and A. MADRY. 2019, «Exploring the landscape of spatial robustness», in *International Conference on Machine Learning*, PMLR, p. 1802–1811. 196, 204

FEINMAN, R., R. R. CURTIN, S. SHINTRE and A. B. GARDNER. 2017, «Detecting adversarial samples from artifacts», *arXiv preprint arXiv:1703.00410*. 193, 194, 197

GAO, R., Q. GUO, F. JUEFEI-XU, H. YU and W. FENG. 2021, «Advhaze: Adversarial haze attack», *arXiv preprint arXiv:2104.13673*. 193

GOODFELLOW, I. J., J. SHLENS and C. SZEGEDY. 2014, «Explaining and harnessing adversarial examples», *arXiv preprint arXiv:1412.6572*. 195, 201

GROSSE, K., P. MANOHARAN, N. PAPERNOT, M. BACKES and P. D. MCDANIEL. 2017, «On the (statistical) detection of adversarial examples», *arXiv preprint arXiv:1702.06280*. 197

GUO, C., G. PLEISS, Y. SUN and K. Q. WEINBERGER. 2017, «On calibration of modern neural networks», in *Proceedings of the 34th International Conference on Machine Learning, ICML*, vol. 70, p. 1321–1330. 193

HALLIN, M., D. PAINDAVEINE and M. ŠIMAN. 2010, «Multivariate quantiles and multiple-output regression quantiles: From l1 optimization to halfspace depth», *The Annals of Statistics*, vol. 38, nº 2, p. 635–669. 198

HE, K., X. ZHANG, S. REN and J. SUN. 2016, «Deep residual learning for image recognition», in *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 770–778. 194

HE, L., Q. ZHOU, X. LI, L. NIU, G. CHENG, X. LI, W. LIU, Y. TONG, L. MA and L. ZHANG. 2021, «End-to-end video object detection with spatial-temporal transformers», in *Proceedings of the 29th ACM International Conference on Multimedia*, p. 1507–1516. 194

HENDRYCKS, D., K. LEE and M. MAZEIKA. 2019, «Using pre-training can improve model robustness and uncertainty», in *International Conference on Machine Learning*, PMLR, p. 2712–2721. 196

HUANG, L., C. ZHANG and H. ZHANG. 2020, «Self-adaptive training: beyond empirical risk minimization», *Advances in Neural Information Processing Systems*, vol. 33. 196

JIA, Y. J., Y. LU, J. SHEN, Q. A. CHEN, H. CHEN, Z. ZHONG and T. W. WEI. 2020, «Fooling detection alone is not enough: Adversarial attack against multiple object tracking», in *International Conference on Learning Representations (ICLR'20)*. 193

JIAO, X., Y. YIN, L. SHANG, X. JIANG, X. CHEN, L. LI, F. WANG and Q. LIU. 2019, «Tinybert: Distilling bert for natural language understanding», *arXiv preprint arXiv:1909.10351*. 200

JOHNSON, C. 2018, «The increasing risks of risk assessment: On the rise of artificial intelligence and non-determinism in safety-critical systems», in *the 26th Safety-Critical Systems Symposium*, Safety-Critical Systems Club York, UK., p. 15. 192

KHERCHOUCHE, A., S. A. FEZZA, W. HAMIDOUCHE and O. DÉFORGES. 2020, «Natural scene statistics for detecting adversarial examples in deep neural networks», in *22nd IEEE International Workshop on Multimedia Signal Processing*, IEEE, p. 1–6. 193, 194, 197

KOSHEVOY, G. and K. MOSLER. 1997, «Zonoid trimming for multivariate distributions», *The Annals of Statistics*, vol. 25, nº 5, p. 1998–2017. 198

KRIZHEVSKY, A. 2009, «Learning multiple layers of features from tiny images», cahier de recherche. 200

KURAKIN, A., I. J. GOODFELLOW and S. BENGIO. 2018, «Adversarial examples in the physical world», in *Artificial intelligence safety and security*, Chapman and Hall/CRC, p. 99–112. 195

LEIBIG, C., V. ALLKEN, M. S. AYHAN, P. BERENS and S. WAHL. 2017, «Leveraging uncertainty information from deep neural networks for disease detection», *Scientific reports*, vol. 7, nº 1, p. 1–14. 193

LI, X. and F. LI. 2017, «Adversarial examples detection in deep networks with convolutional filter statistics», in *IEEE International Conference on Computer Vision, ICCV*, IEEE Computer Society, p. 5775–5783. 197

LIANG, B., H. LI, M. SU, X. LI, W. SHI and X. WANG. 2021, «Detecting adversarial image examples in deep neural networks with adaptive noise reduction», *IEEE Trans. Dependable Secur. Comput.*, vol. 18, nº 1, p. 72–85. 197

LIN, J., C. SONG, K. HE, L. WANG and J. E. HOPCROFT. 2019, «Nesterov accelerated gradient and scale invariance for adversarial attacks», *arXiv preprint arXiv:1908.06281*. 193

LIU, F. T., K. M. TING and Z.-H. ZHOU. 2008, «Isolation forest», in *2008 eighth ieee international conference on data mining*, IEEE, p. 413–422. 193

LIU, R. Y. 1990, «On a notion of data depth based on random simplices», *The Annals of Statistics*, vol. 18, nº 1, p. 405–414. 198

LIU, R. Y. 1992, *Data Depth and Multivariate Rank Tests*, North-Holland, Amsterdam, p. 279–294. 198

LU, J., T. ISSARANON and D. A. FORSYTH. 2017, «Safetynet: Detecting and rejecting adversarial examples robustly», in *IEEE International Conference on Computer Vision*, IEEE Computer Society, p. 446–454. 197

MA, S. and Y. LIU. 2019, «Nic: Detecting adversarial samples with neural network invariant checking», in *Proceedings of the 26th network and distributed system security symposium (NDSS 2019)*. 199

MA, S., Y. LIU, G. TAO, W. LEE and X. ZHANG. 2019, «NIC: detecting adversarial samples with neural network invariant checking», in *26th Annual Network and Distributed System Security Symposium*, The Internet Society. 197

MA, X., B. LI, Y. WANG, S. M. ERFANI, S. N. R. WIJEWICKREMA, G. SCHOENEBECK, D. SONG, M. E. HOULE and J. BAILEY. 2018, «Characterizing adversarial subspaces using local intrinsic dimensionality», in *6th International Conference on Learning Representations*. 193, 194, 197

MADRY, A., A. MAKELOV, L. SCHMIDT, D. TSIPRAS and A. VLADU. 2018, «Towards deep learning models resistant to adversarial attacks», in *International Conference on Learning Representations*. 196, 201

MAHMOOD, K., R. MAHMOOD and M. VAN DIJK. 2021, «On the robustness of vision transformers to adversarial examples», in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, p. 7838–7847. 196

MEINKE, A. and M. HEIN. 2020, «Neural networks that provably know when they don't know», in *8th International Conference on Learning Representations, ICLR*. 193

MENG, D. and H. CHEN. 2017, «Magnet: A two-pronged defense against adversarial examples», in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, édité par B. M. Thuraisingham, D. Evans, T. Malkin and D. Xu, ACM, p. 135–147. 193, 194, 197, 199, 201

METZEN, J. H., T. GENEWEIN, V. FISCHER and B. BISCHOFF. 2017, «On detecting adversarial perturbations», in *5th International Conference on Learning Representations*. 197

MOOSAVI-DEZFOOLI, S.-M., A. FAWZI and P. FROSSARD. 2016, «Deepfool: a simple and accurate method to fool deep neural networks», in *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 2574–2582. 195, 201

MORGULIS, N., A. KREINES, S. MENDELOWITZ and Y. WEISGLASS. 2019, «Fooling a real car with adversarial traffic signs», *arXiv preprint arXiv:1907.00374*. 206

MOZHAROVSKYI, P., K. MOSLER and T. LANGE. 2015, «Classifying real-world data with the DDα-procedure», *Advances in Data Analysis and Classification*, vol. 9, nº 3, p. 287–314. 198

NASEER, M., S. KHAN, M. HAYAT, F. S. KHAN and F. PORIKLI. 2021, «On generating transferable targeted perturbations», in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, p. 7708–7717. 193

PANG, T., C. DU, Y. DONG and J. ZHU. 2018, «Towards robust detection of adversarial examples», in *Advances in Neural Information Processing Systems 31*, p. 4584–4594. 196

PARMAR, N., A. VASWANI, J. USZKOREIT, L. KAISER, N. SHAZEER, A. KU and D. TRAN. 2018, «Image transformer», in *International Conference on Machine Learning*, PMLR, p. 4055–4064. 194

RAGHURAM, J., V. CHANDRASEKARAN, S. JHA and S. BANERJEE. 2021, «A general framework for detecting anomalous inputs to dnn classifiers», in *International Conference on Machine Learning*, PMLR, p. 8764–8775. 197

RAMSAY, K., S. DUROCHER and A. LEBLANC. 2019, «Integrated rank-weighted depth», *Journal of Multivariate Analysis*, vol. 173, p. 51–69. 198

RICE, L., E. WONG and Z. KOLTER. 2020, «Overfitting in adversarially robust deep learning», in *International Conference on Machine Learning*, PMLR, p. 8093–8104. 196

ROUSSEEUW, P. J. and M. HUBERT. 1999, «Regression depth», *Journal of the American Statistical Association*, vol. 94, nº 446, p. 388–402. 198

ROUSSEEUW, P. J. and M. HUBERT. 2018, «Anomaly detection by robust statistics», *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, nº 2, p. e1236. 198

ROUSSEEUW, P. J. and A. STRUYF. 1998, «Computing location depth and regression depth in higher dimensions», *Statistics and Computing*, vol. 8, nº 3, p. 193–203. 198

SCHÖLKOPF, B., J. PLATT, J. SHAWE-TAYLOR, A. SMOLA and R. WILLIAMSON. 2001, «Estimating the support of a high-dimensional distribution», *Neural Computation*, vol. 13, nº 7, p. 1443–1471. 193

SERFLING, R. 2006, «Depth functions in nonparametric multivariate inference», *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 72. 198

SHE, Y., S. TANG and J. LIU. 2021, «On generalization and computation of tukey's depth: Part i», *arXiv preprint arXiv:2112.08475*. 204

SOTGIU, A., A. DEMONTIS, M. MELIS, B. BIGGIO, G. FUMERA, X. FENG and F. ROLI. 2020, «Deep neural rejection against adversarial examples», *EURASIP J. Inf. Secur.*, vol. 2020, p. 5. 196, 197

STAERMAN, G. 2022, *Functional anomaly detection and robust estimation*, thèse de doctorat, Institut polytechnique de Paris. 198

STAERMAN, G., E. ADJAKOSSA, P. MOZHAROVSKYI, V. HOFER, J. S. GUPTA and S. CLÉMENÇON. 2022, «Functional anomaly detection: a benchmark study», *arXiv preprint arXiv:2201.05115*. 198

STAERMAN, G., P. MOZHAROVSKYI, S. CLÉMENÇON and F. D'ALCHÉ BUC. 2019, «Functional isolation forest», in *Proceedings of The Eleventh Asian Conference on Machine Learning*, vol. 101, p. 332–347. 193

STAERMAN, G., P. MOZHAROVSKYI and S. CLÉMENÇON. 2020, «The area of the convex hull of sampled curves: a robust functional statistical depth measure», in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, vol. 108, PMLR, p. 570–579. 193, 198

STAERMAN, G., P. MOZHAROVSKYI and S. CLÉMENÇON. 2021a, «Affine-invariant integrated rank-weighted depth: Definition, properties and finite sample analysis», *arXiv preprint arXiv:2106.11068*. 198

STAERMAN, G., P. MOZHAROVSKYI, P. COLOMBO, S. CLÉMENÇON and F. D'ALCHÉ BUC. 2021b, «A pseudo-metric between probability distributions based on depth-trimmed regions», *arXiv preprint arXiv:2103.12711*. 198

STEINER, A., A. KOLESNIKOV, , X. ZHAI, R. WIGHTMAN, J. USZKOREIT and L. BEYER. 2021, «How to train your vit? data, augmentation, and regularization in vision transformers», *arXiv preprint arXiv:2106.10270*. 194

SUBBASWAMY, A. and S. SARIA. 2020, «From development to deployment: dataset shift, causality, and shift-stable models in health ai», *Biostatistics*, vol. 21, nº 2, doi: 10.1093/biostatistics/kxz041, p. 345–352, ISSN 1465-4644. 192

SZEGEDY, C., W. ZAREMBA, I. SUTSKEVER, J. BRUNA, D. ERHAN, I. J. GOODFELLOW and R. FERGUS. 2014, «Intriguing properties of neural networks», in *2nd International Conference on Learning Representations*. 193, 195

TOLSTIKHIN, I., N. HOULSBY, A. KOLESNIKOV, L. BEYER, X. ZHAI, T. UNTERTHINER, J. YUNG, A. STEINER, D. KEYSERS, J. USZKOREIT, M. LUCIC and A. DOSOVITSKIY. 2021, «Mlp-mixer: An all-mlp architecture for vision», *arXiv preprint arXiv:2105.01601*. 194

TRAMER, F., N. CARLINI, W. BRENDEL and A. MADRY. 2020, «On adaptive attacks to adversarial example defenses», *Advances in Neural Information Processing Systems*, vol. 33, p. 1633–1645. 204

TUKEY, J. W. 1975, «Mathematics and the picturing of data», in *Proceedings of the International Congress of Mathematicians*, vol. 2, p. 523–531. 193, 197, 198

WANG, X., X. HE, J. WANG and K. HE. 2021a, «Admix: Enhancing the transferability of adversarial attacks», in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, p. 16 158–16 167. 193

WANG, Y., R. HUANG, S. SONG, Z. HUANG and G. HUANG. 2021b, «Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition», *Advances in Neural Information Processing Systems*, vol. 34. 194

WANG, Y., D. ZOU, J. YI, J. BAILEY, X. MA and Q. GU. 2019, «Improving adversarial robustness requires revisiting misclassified examples», in *International Conference on Learning Representations*. 196

WU, D., S.-T. XIA and Y. WANG. 2020a, «Adversarial weight perturbation helps robust generalization», *Advances in Neural Information Processing Systems*, vol. 33. 196

WU, K., A. WANG and Y. YU. 2020b, «Stronger and faster wasserstein adversarial attacks», in *International Conference on Machine Learning*, PMLR, p. 10 377–10 387. 193

XIE, Y., D. WANG, P.-Y. CHEN, J. XIONG, S. LIU and O. KOYEJO. 2022, «A word is worth a thousand dollars: Adversarial attack on tweets fools stock prediction», in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Seattle, United States, p. 587–599. 206

XU, W., D. EVANS and Y. QI. 2018, «Feature squeezing: Detecting adversarial examples in deep neural networks», in *25th Annual Network and Distributed System Security Symposium*, The Internet Society. 193, 194, 197, 199, 201

ZHAI, X., X. WANG, B. MUSTAFA, A. STEINER, D. KEYSERS, A. KOLESNIKOV and L. BEYER. 2022, «Lit: Zero-shot transfer with locked-image text tuning», *CVPR*. 194

ZHANG, H., Y. YU, J. JIAO, E. P. XING, L. E. GHAOUI and M. I. JORDAN. 2019, «Theoretically principled trade-off between robustness and accuracy», in *International Conference on Machine Learning*, p. 1–11. 196, 201

ZHAO, Z., Z. LIU and M. LARSON. 2020, «Adversarial color enhancement: Generating unrestricted adversarial images by optimizing a color filter», *arXiv preprint arXiv:2002.01008*. 193

ZHENG, Z. and P. HONG. 2018, «Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks», in *Advances in Neural Information Processing Systems 31*, p. 7924–7933. 197

ZUO, Y. and R. SERFLING. 2000, «General notions of statistical depth function», *The Annals of Statistics*, vol. 28, nº 2, p. 461–482. 193, 198

Part II Conclusion

Part II focused on the second research question: how can we craft an efficient and effective detection method based on simple tools? This research question has been addressed in two different ways:

1. **In the supervised case.** We tackled the problem of crafting a detection method to distinguish between natural and attacked samples. We decided to use the data-depths, simple statistical tools providing a center-outward ordering of points w.r.t. to a reference distribution. We leveraged the class-wise information present in the model's different layers to protect and the knowledge about the potential threats to craft an efficient detector. We experimentally showed the superiority of our proposed method.

2. **In the unsupervised scenario.** We also proposed a method to discard attacks in the case where no information about the threats is available. We made use of the information extracted by the data-depths as well as the internal structure of the Vision Transformer, which behaves differently than the classically used ResNet, to construct an efficient method to detect adversarial examples. Experimentally, we proved that our method outperforms other state-of-the-art detectors and complicates the attacker's job.

Part II was dedicated to increasing our trust in the deep models' inputs by ensuring they come from a clean environment.
In Section 7.6, we will provide the reader with concluding remarks about the work we produced during the last four years. We will also put our contributions into perspective and discuss some of the limitations as well as the perspectives of the different parts of our work.

Chapter 8

Discussion of Findings, Limitations, Potential Future Works and Global Summary

Contents

This manuscript presented the research conducted over the last four years on improving the reliability of neural networks under attack. In this chapter, we will provide some concluding remarks.

8.1 Discussion of Findings

8.1.1 Research problem

In this thesis, the work presented focused on two central axes.

In the first axis, we focused on answering the following research question: (**Q1**) “How can we use the internal structure of DNN’s output space to improve its robustness?".

In the second axis, we focused on providing answers to the following research question: (**Q2**) “How can we craft an efficient and effective detection method based on simple tools".

In the following, we will briefly summarize our key results.

8.1.2 Key results

*Our proposed solution for (**Q1**).*

In Part I, we present our solution to the first research question for two different applications: computer vision and Smart Grid systems. We proposed a method based on the regularization of the training loss by an information-geometric measure, namely the Fisher-Rao distance, which computes the geodesic distance between two probability distributions on the underlying probabilistic manifold that the outputs of a DNN form. We connected it to standard metrics and experimentally verified that using the Fisher-Rao distance as a regularizer achieved more Pareto-optimum points than the standard regularization measure. On image data, we experimentally proved that our method consistently improved the DNNs robustness to adversarial attacks. We later adapted our method in the context of the Smart Grids. Due to monitoring and service necessity, intelligent power grids must rely on cyber-component, making them vulnerable to cyber-attacks. State estimators, necessary components to monitor the grid, are known to be particularly sensitive to stealth attacks. During this work, we presented a DNN-based, more specifically a Variational AutoEncoder-based, solution to improve the robustness of state estimators in the context of Smart Grid systems to stealth attacks.

*Our proposed solution for (**Q2**).*

In Part II, we presented two solutions to try and answer the second research question suited to two different scenarios: the supervised and unsupervised cases. In both cases, we used the data-depths, simple statistical tools that provide a center-outward ordering of a point with respect to a reference distribution. For a given new input, the data-depth will measure how deep it is in the reference distribution. In the supervised case, we used a specific depth, the halfspace-mass depth, and leveraged the class-wise information contained in the different layers of a given model, to

craft an efficient detection method. We experimentally showed that our method consistently improved the state-of-the-art method on different kinds of attacks. In the unsupervised case, we used another data-depth, called the Integrated Rank-Weighted depth, and leveraged the particular structure of Vision Transformers to build an efficient method to detect adversarial samples. Experimentally, in addition to proving the superiority of our approach compared to other state-of-the-art methods, we showed that using our method would toughen the attacker's job at attacking both the targeted classifier and the defense.

8.1.3 Comparison between our work and previous state-of-the-art methods.

Globally, our proposed methods show an improvement compared to previous state-of-the-art methods.

Using the Fisher-Rao distance to improve defenses. Regarding robustness as a defense against adversarial attacks, theoretically, we have shown that the Kullback-Leibler divergence is a surrogate of the Fisher-Rao distance. In addition, compared to the use of the Kullback-Leibler divergence [ZHANG and collab., 2019] as a robust regularizer, using the Fisher-Rao distance to force clean and adversarial predictions to be similar experimentally suggests that it is possible to achieve better trade-offs between natural and adversarial accuracies, both on simple data and on more realistic ones. Finally, as for the Kullback-Leibler divergence, combining the Fisher-Rao regularizer with other methods to improve robustness is possible.

Protecting against False Data Injection Attacks in Smart Grid systems. In the case of building robust state estimators to enhance the protection of Smart Grids systems, using a Variational AutoEncoder, combined with the online generation of false data injection attacks (FDIAs) [LIU and collab., 2011] and our proposed robust loss, is allowing us to build strong state estimators, with good robust qualities. In the case of DC assumptions, our method can rightfully estimate the state even in the case of an attack. In the case of AC assumptions, our method clearly outperforms the LASSO solution proposed by JIN and collab. [2019], which is one the most recently proposed method to improve state estimators' defenses against FDIAs.

Using data-depths to detect adversarial examples. Contrary to other state-of-the-art methods, which are based on heavy ad-hoc or training, we decided to use a simple statistical tool named the data-depth to build detection methods. Data-depths have three main advantages compared to other methods: they are fast to compute, have a simple geometric interpretation since they provide a center-outward

ordering of a new input compared to a reference probability distribution, and are non-differentiable, making the attacker's job more difficult. In the case of supervised methods, where the defense has either complete or partial knowledge about the threats it is going to face, we proposed HAMPER, a supervised method that leverages the halfspace-mass depth score at different layer levels. We presented two versions of our solution, one where complete knowledge about the attacker is available and another where only partial knowledge is available. Our method is somehow similar to the Local Intrinsic Dimensionality method as we compare a new point to a reference. However, we only compute our anomality score on a subset of layers and not all of them. Experimentally, our proposed detection method significantly and consistently outperforms previous state-of-the-art methods in all considered scenarios, namely the attack-aware scenario, the blind-to-attack scenario, or under adaptive attacks. In terms of computational time, our method is comparable to the fastest methods, namely the Natural Scene Statistics method [KHERCHOUCHE and collab., 2020] and the Kernel Density and Bayesian Uncertainty method [FEINMAN and collab., 2017]. In the unsupervised case, where the defense does not have any knowledge about the possible threats it is going to face, our proposed data-depth method named APPROVED significantly outperforms other state-of-the-art methods, both in terms of performances under classical and adaptive attacks, and in terms of computational requirements. While there are still improvements to provide in all cases, our different experiments suggest that data-depths are valuable tools to distinguish between natural and attacked samples.

8.2 Limitations and Potential Future Works

This section will put our work in perspective and mention some potential future work.

8.2.1 Limitations of our proposed methods

Using the Fisher-Rao distance to improve defenses. While the Fisher-Rao distance has a simple geometric interpretation, and its use over the Kullback-Leibler divergence as a regularizer improves the trade-off between natural and robust performances of deep classifiers, we do not have theoretical guarantees over the performances. In addition, while deeply connected, our experiments suggest that the Fisher-Rao distance and the Kullback-Leibler divergence behave differently. A theoretical explanation of this behavior change could be interesting.

Protecting state estimators in Smart Grid systems. We proposed methods to protect state estimators against attacks. While our results suggest that our method is

efficient, we could not test it on real-world data since they were unavailable.

Building strong detection methods. While significantly improving the state-of-the-art results on multiple threats, the proposed methods are not perfect. Adversarial attacks, while representing the worst-case scenario, are not the only possible alteration of the input. However, we do not know if our proposed methods can detect all potential threats in all possible scenarios, as we focused solely on protecting deep classifiers against adversarial attacks.

8.2.2 Potential future work related to: How can we use the internal structure of DNN's output space to improve its robustness?

- **Extention of our method to multimodal models.** The will of the community to build multimodal systems, which learn from more than one type of modality, typically combining text and image data, has gained importance over the last few months [ALAYRAC and collab., 2022; GARCIA and collab., 2019; RADFORD and collab., 2021; RAMESH and collab., 2022, 2021]. Recently, methods to attack such systems have been developed [EVTIMOV and collab., 2020; YI and collab., 2021; ZHANG and collab., 2022]. Protecting those networks will therefore be necessary when these models are deployed in practical scenarios. However, protecting such systems is not trivial, as textual data is discrete and based on a dictionary while image data is not, therefore, we need to ensure that all modalities are taken into account and protected.

- **Protecting Vision Transformers.** Vision Transformers (ViTs) are achieving state-of-the-art performances on multiple vision tasks today [bey; FAYYAZ and collab., 2021; GRAHAM and collab., 2021; HEO and collab., 2021; LEE and collab., 2021; LI and collab., 2021, 2022; MEHTA and RASTEGARI, 2021; RENGGLI and collab., 2022; SANDLER and collab., 2022; TOUVRON and collab., 2022, 2021; TU and collab., 2022; YANG and collab., 2022]. However, it has been widely overlooked by the robustness community, just a few works have been studying whether ViT where, by design, more robust than classic convolutional networks [ALDAHDOOH and collab., 2021a; BENZ and collab., 2021; MAHMOOD and collab., 2021]. It would therefore be interesting to adapt our method to improve the robustness of Vision Transformers on multiple vision tasks. This task in not trivial as Vision Transformers are a huge amount of parameters and their training on natural samples is already heavy.

- **Additional theoretical comparison between Fisher-Rao distance and other metrics.** We have shown, in Chapter 3 using a toy example, that the optimization based on FIRE is well-behaved and gives all the desired Pareto-optimal

points in the natural-adversarial region. This observation contrasts with the results of other state-of-the-art adversarial learning approaches. Further theoretical explanations of this behavior change would be worth exploring.

8.2.3 Potential future work related to: How can we craft an efficient and effective detection method based on simple tools?

- **Extention to different underlying tasks.** Our proposed methods can be extended to any underlying tasks. Attacks have started to be designed for a wide range of vision tasks, such as image segmentation [HENDRIK METZEN and collab., 2017; XIE and collab., 2017], object recognition [XIE and collab., 2017], speech recognition [CISSE and collab., 2017]. It, therefore, would be interesting to try to protect these underlying tasks. Moreover, other fields have protection necessities such as securing textual pre-trained models [CHAPUIS and collab., 2020; COLOMBO and collab., 2021a; DEVLIN and collab., 2018; DINKAR and collab., 2020; YANG and collab., 2019], which remains in its infancy. In NLP, many opportunities exist where attacks could help quantify model robustness. Fields of interest include sequence generation task COLOMBO and collab. [2021c, 2022d]; COLOMBO* and collab. [2019]; COLOMBO and collab. [2021e]; JALALZAI* and collab. [2020], translation models COLOMBO and collab. [2021a]; GUERREIRO and collab. [2023], multimodal emotion analysis COLOMBO and collab. [2022d]; GARCIA and collab. [2019], and sentence similarity CHHUN and collab. [2022]; COLOMBO and collab. [2021b, 2022a,c, 2021d]; STAERMAN and collab. [2021] as they are widely used and often deployed in real-world scenarios. As each architecture and each application presents its specificities, adapting our proposed methods to other tasks is not straightforward.

- **Using information tools to protect systems.** As shown in Part II, the data-depth is an interesting tool to distinguish between normal and modified inputs. However, other metrics could be well suited for this task COLOMBO [2021a]; COLOMBO and collab. [2022b]; PICHLER and collab. [2022]. Information measures compute the resemblance between two probability distributions and are gaining attention in the deep learning community. It would be interesting to find out if such measures can help protect our systems.

- **Combining detection methods.** We know that the different state-of-the-art methods and our methods capture different characteristics of clean and adversarial inputs, as they tend to behave differently depending on the type of threats they face. It would be interesting to find a simple method to combine the scores provided by different methods to leverage the different characteristics of natural and adversarial input instead of focusing on only one.

- **Real time detector.** To deploy detection methods in real-world scenarios, we believe they need to meet three separate requirements.

  (R1) **Black-box scenario.** Systems already deployed in production are generally opaque to the end user, who only has access to the softmax predictions of the networks.

  (R2) **Low resources / computation time.** In many real-world applications, AI systems are making real-time predictions at a high frequency (*e.g.* face recognition for airport security). As a result, any relevant detector should have a low inference time and require low computation resources.

  (R3) **No oracle on the nature of the attackers.** Any relevant detector should be *unsupervized*, meaning it should not require any training phase with access to attack examples. Indeed, the landscape of existing attackers is moving fast, making the availability of adversarial examples not realistic in practice.

  Existing detection methods tend to fail at meeting at least one of those requirements.

  Supervised methods – not satisfying (R3). Supervised methods usually consist in training simple machine learning algorithms, such as SVMs or logistic regressions, to discriminate adversarial examples from natural ones, using examples from both classes. The features used for training these machine learning models can be extracted from the network's layers using the samples directly [CARRARA and collab., 2018; COLOMBO, 2021b; LU and collab., 2017; METZEN and collab., 2017], or pre-process them using kernel density estimation or uncertainty measure [FEINMAN and collab., 2017], computer vision specific characteristics such as natural scene statistics [KHERCHOUCHE and collab., 2020], PCA [LI and LI, 2017] or also local intrinsic dimensionality [MA and collab., 2018]. Regarding **(R2)**, one can arguably say that these methods are satisfying as the inference time of simple machine learning models is fast. However, as they do not satisfy **(R3)**, these methods need to make some assumptions on the nature of adversarial attacks to generate malicious samples at the risk of overfitting and misgeneralizing. Moreover, most of these methods rely on the hidden layers of the networks, which makes them unrealistic for practical black-box applications **(R1)** where only softmax are available.

  Unsupersived methods – satisfying (R3). Unsupervised methods only rely on clean samples to build a detector, making them very attractive for real-life applications. Let us explore existing works in light of requirements **(R1)** and **(R2)**. Some detectors require access to intermediate layers representation [ALDAHDOOH and collab., 2021b; MA and collab., 2019; SOTGIU and collab., 2020;

ZHENG and HONG, 2018], which makes them unsuitable for use in the context of **(R1)**. Two methods satisfy **(R1)** but are arguably less effective regarding **(R2)**: the Feature Squeezing (FS) of XU and collab. [2018] and the MagNet detector of MENG and CHEN [2017] which relies on a denoising autoencoder. Let us also mention JTLA [RAGHURAM and collab., 2021], a refinement of FS which unfortunately does not satisfy **(R1)**.

Table 8.1: Summary of Detector's requirements meets

Detector	(R1)	(R2)	(R3)
MA and collab. [2019]	✗	✗	✓
SOTGIU and collab. [2020]	✗	✗	✓
ZHENG and HONG [2018]	✗	✗	✓
ALDAHDOOH and collab. [2021b]	✗	✗	✓
XU and collab. [2018]	✓	✗	✓
MENG and CHEN [2017]	✓	✗	✓
RAGHURAM and collab. [2021]	✗	✗	✓

Out-Of-Distribution detection methods. As adversarial attack detection can be considered an extreme case of the out-of-distribution (OOD) detection problem, it would be interesting to extend ODD methods to adversarial detection. In particular, a line of OOD methods is based on extracting relevant information from the softmax probabilities, making them very attractive as they meet all the requirements. This line of work has been launched by the seminal work of DARRIN and collab. [2023a,b]; GOMES and collab.; HENDRYCKS and GIMPEL [2016], who proposed to focus on the Maximum Softmax Probability (MSP) to discriminate between in- and out-of-distribution samples. The underlying idea of MSP is that the more spiky the probabilities, the more confident the network is and, therefore, the cleaner the input. Let us also mention the DOCTOR detector, recently introduced by GRANESE and collab. [2021], which computes the Gini coefficient of the softmax probabilities. Both methods satisfy the three requirements **(R1)** - **(R2)** - **(R3)**, but fail at detecting some attacks.

8.2.4 Global research directions

Adapting the adversarial problem to the real world. The adversarial problem represents a worst-case scenario and is an exciting problem to solve. However, this scenario is not necessarily representative of what could happen in real-world applications. For example, providing full knowledge about a system to an attacker is probably too ambitious. In addition, defenses can be combined, i.e., it is possible to craft a detector on top of a robust network. Finally, defending directly on the neural network to protect is not the only type of defense that exists. For example, one can monitor the activities of its users to ensure that no strange behavior is happening, such as an abnormal number of queries in a short amount of time. While the developed tools for the adversarial problem are powerful and interesting and give us insights into the functioning of neural networks, there is still a lot to do to craft efficient defensive methods that can be applied to the industry in more practical

cases.

Are the attack and defense problems underspecified? Recently, the problem of underspecification and its implications has been formalized by D'AMOUR and collab. [2020]; TENEY and collab. [2022]. A problem is considered underspecified if there exist many different solutions that solve it without any change of results on the main task but exhibiting widely different characteristics. Most DL problems are, in fact, underspecified, and it could cause problems with the reliability of machine learning, as every solution can behave differently when deployed in real-world applications where the main task can slightly vary from the training one. We believe the attack problem is, in fact, underspecified, as attacking two different models trained on the same task can result in vastly different attacked samples. Moreover, we also believe the defense problem is underspecified, as the performances of the defenses on attacks that have never been seen before can widely change depending on the chosen defense [D'AMOUR and collab., 2020][Section 5]. We, therefore, believe it would be interesting to test attack and defense mechanisms under the scope of underspecification.

8.3 References

221

ALAYRAC, J.-B., J. DONAHUE, P. LUC, A. MIECH, I. BARR, Y. HASSON, K. LENC, A. MENSCH, K. MILLICAN, M. REYNOLDS and collab.. 2022, «Flamingo: a visual language model for few-shot learning», *arXiv preprint arXiv:2204.14198*. 221

ALDAHDOOH, A., W. HAMIDOUCHE and O. DEFORGES. 2021a, «Reveal of vision transformers robustness against adversarial attacks», *arXiv preprint arXiv:2106.03734*. 221

ALDAHDOOH, A., W. HAMIDOUCHE and O. DÉFORGES. 2021b, «Revisiting model's uncertainty and confidences for adversarial example detection», *arXiv preprint arXiv: 2103.05354*. 223, 224

BENZ, P., S. HAM, C. ZHANG, A. KARJAUV and I. S. KWEON. 2021, «Adversarial robustness comparison of vision transformer and mlp-mixer to cnns», *arXiv preprint arXiv:2110.02797*. 221

CARRARA, F., R. BECARELLI, R. CALDELLI, F. FALCHI and G. AMATO. 2018, «Adversarial examples detection in features distance spaces», in *Computer Vision - ECCV 2018 Workshops - Munich, Germany, Proceedings, Part II*, vol. 11130, Springer, p. 313–327. 223

CHAPUIS, E., P. COLOMBO, M. MANICA, M. LABEAU and C. CLAVEL. 2020, «Hierarchical pre-training for sequence labelling in spoken dialog», *arXiv preprint arXiv:2009.11152*. 222

CHHUN, C., P. COLOMBO, C. CLAVEL and F. M. SUCHANEK. 2022, «Of human criteria and automatic metrics: A benchmark of the evaluation of story generation», *(oral) COLING 2022*. 222

CISSE, M. M., Y. ADI, N. NEVEROVA and J. KESHET. 2017, «Houdini: Fooling deep structured visual and speech recognition models with adversarial examples», *Advances in neural information processing systems*, vol. 30. 222

COLOMBO, P. 2021a, *Apprendre à représenter et à générer du texte en utilisant des mesures d'information*, thèse de doctorat, (PhD thesis) Institut Polytechnique de Paris. 222

COLOMBO, P. 2021b, *Learning to represent and generate text using information measures*, thèse de doctorat, (PhD thesis) Institut polytechnique de Paris. 223

COLOMBO, P., E. CHAPUIS, M. LABEAU and C. CLAVEL. 2021a, «Code-switched inspired losses for spoken dialog representations», in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 8320–8337. 222

COLOMBO, P., C. CLAVE and P. PIANTANIDA. 2021b, «Infolm: A new metric to evaluate summarization & data2text generation», *(best student paper award) AAAI 2022*. 222

COLOMBO, P., C. CLAVEL and P. PIANTANIDA. 2021c, «A novel estimator of mutual information for learning to disentangle textual representations», *(oral) ACL 2021*. 222

COLOMBO, P., E. D. GOMES, G. STAERMAN, N. NOIRY and P. PIANTANIDA. 2022a, «Beyond mahalanobis-based scores for textual ood detection», *arXiv preprint arXiv:2211.13527*. 222

COLOMBO, P., N. NOIRY, E. IRUROZKI and S. CLÉMENÇON. 2022b, «What are the best systems? new perspectives on nlp benchmarking», *NeurIPS 2022*. 222

COLOMBO, P., M. PEYRARD, N. NOIRY, R. WEST and P. PIANTANIDA. 2022c, «The glass ceiling of automatic evaluation in natural language generation», *arXiv preprint arXiv:2208.14585*. 222

COLOMBO, P., G. STAERMAN, C. CLAVEL and P. PIANTANIDA. 2021d, «Automatic text evaluation through the lens of wasserstein barycenters», *(oral) EMNLP 2021*. 222

COLOMBO, P., G. STAERMAN, N. NOIRY and P. PIANTANIDA. 2022d, «Learning disentangled textual representations via statistical measures of similarity», *(oral) ACL 2022*. 222

COLOMBO*, P., W. WITON*, A. MODI, J. KENNEDY and M. KAPADIA. 2019, «Affect-driven dialog generation», *NAACL 2019*. 222

COLOMBO, P., C. YANG, G. VARNI and C. CLAVEL. 2021e, «Beam search with bidirectional strategies for neural response generation», *ICNLSP 2021*. 222

DARRIN, M., P. PIANTANIDA and P. COLOMBO. 2023a, «Rainproof: An umbrella to shield text generators from out-of-distribution data», *arXiv preprint arXiv:2212.09171*. 224

DARRIN, M., G. STAERMAN, E. D. C. GOMES, J. C. CHEUNG, P. PIANTANIDA and P. COLOMBO. 2023b, «Unsupervised layer-wise score aggregation for textual ood detection», *arXiv preprint arXiv:2302.09852*. 224

DEVLIN, J., M.-W. CHANG, K. LEE and K. TOUTANOVA. 2018, «Bert: Pre-training of deep bidirectional transformers for language understanding», *arXiv preprint arXiv:1810.04805*. 222

DINKAR, T., P. COLOMBO, M. LABEAU and C. CLAVEL. 2020, «The importance of fillers for text representations of speech transcripts», *arXiv preprint arXiv:2009.11340*. 222

D'AMOUR, A., K. HELLER, D. MOLDOVAN, B. ADLAM, B. ALIPANAHI, A. BEUTEL, C. CHEN, J. DEATON, J. EISENSTEIN, M. D. HOFFMAN and collab.. 2020, «Underspecification presents challenges for credibility in modern machine learning», *Journal of Machine Learning Research*. 225

EVTIMOV, I., R. HOWES, B. DOLHANSKY, H. FIROOZ and C. C. FERRER. 2020, «Adversarial evaluation of multimodal models under realistic gray box assumption», *arXiv preprint arXiv:2011.12902*. 221

FAYYAZ, M., S. A. KOUHPAYEGANI, F. R. JAFARI, E. SOMMERLADE, H. R. V. JOZE, H. PIRSIAVASH and J. GALL. 2021, «Ats: Adaptive token sampling for efficient vision transformers», . 221

FEINMAN, R., R. R. CURTIN, S. SHINTRE and A. B. GARDNER. 2017, «Detecting adversarial samples from artifacts», *arXiv preprint arXiv:1703.00410*. 220, 223

GARCIA, A., P. COLOMBO, S. ESSID, F. D'ALCHÉ BUC and C. CLAVEL. 2019, «From the token to the review: A hierarchical multimodal approach to opinion mining», *arXiv preprint arXiv:1908.11216*. 221, 222

GOMES, E. D. C., P. COLOMBO, G. STAERMAN, N. NOIRY and P. PIANTANIDA. «A functional perspective on multi-layer out-of-distribution detection», . 224

GRAHAM, B., A. EL-NOUBY, H. TOUVRON, P. STOCK, A. JOULIN, H. JÉGOU and M. DOUZE. 2021, «Levit: a vision transformer in convnet's clothing for faster inference», . 221

GRANESE, F., M. ROMANELLI, D. GORLA, C. PALAMIDESSI and P. PIANTANIDA. 2021, «DOCTOR: A simple method for detecting misclassification errors», *arXiv preprint arXiv:2106.02395*. 224

GUERREIRO, N. M., P. COLOMBO, P. PIANTANIDA and A. F. MARTINS. 2023, «Optimal transport for unsupervised hallucination detection in neural machine translation», *arXiv preprint arXiv:2212.09631*. 222

HENDRIK METZEN, J., M. CHAITHANYA KUMAR, T. BROX and V. FISCHER. 2017, «Universal adversarial perturbations against semantic image segmentation», in *Proceedings of the IEEE international conference on computer vision*, p. 2755–2764. 222

HENDRYCKS, D. and K. GIMPEL. 2016, «A baseline for detecting misclassified and out-of-distribution examples in neural networks», *arXiv preprint arXiv:1610.02136*. 224

HEO, B., S. YUN, D. HAN, S. CHUN, J. CHOE and S. J. OH. 2021, «Rethinking spatial dimensions of vision transformers», . 221

JALALZAI*, H., P. COLOMBO*, C. CLAVEL, É. GAUSSIER, G. VARNI, E. VIGNON and A. SABOURIN. 2020, «Heavy-tailed representations, text polarity classification & data augmentation», *NeurIPS 2020*. 222

JIN, M., I. MOLYBOG, R. MOHAMMADI-GHAZI and J. LAVAEI. 2019, «Scalable and robust state estimation from abundant but untrusted data», *IEEE Transactions on Smart Grid*, vol. 11, nº 3, p. 1880–1894. 219

KHERCHOUCHE, A., S. A. FEZZA, W. HAMIDOUCHE and O. DÉFORGES. 2020, «Natural scene statistics for detecting adversarial examples in deep neural networks», in *22nd IEEE International Workshop on Multimedia Signal Processing*, IEEE, p. 1–6. 220, 223

LEE, S. H., S. LEE and B. C. SONG. 2021, «Vision transformer for small-size datasets», . 221

LI, C., J. YANG, P. ZHANG, M. GAO, B. XIAO, X. DAI, L. YUAN and J. GAO. 2021, «Efficient self-supervised vision transformers for representation learning», *ArXiv*, vol. abs/2106.09785. 221

LI, W., X. WANG, X. XIA, J. WU, X. XIAO, M. ZHENG and S. WEN. 2022, «Sepvit: Separable vision transformer», . 221

LI, X. and F. LI. 2017, «Adversarial examples detection in deep networks with convolutional filter statistics», in *IEEE International Conference on Computer Vision, ICCV*, IEEE Computer Society, p. 5775–5783. 223

LIU, Y., P. NING and M. K. REITER. 2011, «False data injection attacks against state estimation in electric power grids», *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, nº 1, p. 1–33. 219

LU, J., T. ISSARANON and D. A. FORSYTH. 2017, «Safetynet: Detecting and rejecting adversarial examples robustly», in *IEEE International Conference on Computer Vision*, IEEE Computer Society, p. 446–454. 223

MA, S., Y. LIU, G. TAO, W. LEE and X. ZHANG. 2019, «NIC: detecting adversarial samples with neural network invariant checking», in *26th Annual Network and Distributed System Security Symposium*, The Internet Society. 223, 224

MA, X., B. LI, Y. WANG, S. M. ERFANI, S. N. R. WIJEWICKREMA, G. SCHOENEBECK, D. SONG, M. E. HOULE and J. BAILEY. 2018, «Characterizing adversarial subspaces using local intrinsic dimensionality», in *6th International Conference on Learning Representations*. 223

MAHMOOD, K., R. MAHMOOD and M. VAN DIJK. 2021, «On the robustness of vision transformers to adversarial examples», in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, p. 7838–7847. 221

MEHTA, S. and M. RASTEGARI. 2021, «Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer», . 221

MENG, D. and H. CHEN. 2017, «Magnet: A two-pronged defense against adversarial examples», in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, édité par B. M. Thuraisingham, D. Evans, T. Malkin and D. Xu, ACM, p. 135–147. 224

METZEN, J. H., T. GENEWEIN, V. FISCHER and B. BISCHOFF. 2017, «On detecting adversarial perturbations», in *5th International Conference on Learning Representations*. 223

PICHLER, G., P. J. A. COLOMBO, M. BOUDIAF, G. KOLIANDER and P. PIANTANIDA. 2022, «A differential entropy estimator for training neural networks», in *(oral) ICML 2022*. 222

RADFORD, A., J. W. KIM, C. HALLACY, A. RAMESH, G. GOH, S. AGARWAL, G. SASTRY, A. ASKELL, P. MISHKIN, J. CLARK and collab.. 2021, «Learning transferable visual models from natural language supervision», in *International Conference on Machine Learning*, PMLR, p. 8748–8763. 221

RAGHURAM, J., V. CHANDRASEKARAN, S. JHA and S. BANERJEE. 2021, «A general framework for detecting anomalous inputs to dnn classifiers», in *International Conference on Machine Learning*, PMLR, p. 8764–8775. 224

RAMESH, A., P. DHARIWAL, A. NICHOL, C. CHU and M. CHEN. 2022, «Hierarchical text-conditional image generation with clip latents», *arXiv preprint arXiv:2204.06125*. 221

RAMESH, A., M. PAVLOV, G. GOH, S. GRAY, C. VOSS, A. RADFORD, M. CHEN and I. SUTSKEVER. 2021, «Zero-shot text-to-image generation», in *International Conference on Machine Learning*, PMLR, p. 8821–8831. 221

RENGGLI, C., A. S. PINTO, N. HOULSBY, B. MUSTAFA, J. PUIGCERVER and C. RIQUELME. 2022, «Learning to merge tokens in vision transformers», . 221

SANDLER, M., A. ZHMOGINOV, M. VLADYMYROV and A. JACKSON. 2022, «Fine-tuning image transformers using learnable memory», . 221

SOTGIU, A., A. DEMONTIS, M. MELIS, B. BIGGIO, G. FUMERA, X. FENG and F. ROLI. 2020, «Deep neural rejection against adversarial examples», *EURASIP J. Inf. Secur.*, vol. 2020, p. 5. 223, 224

STAERMAN, G., P. MOZHAROVSKYI, P. COLOMBO, S. CLÉMENÇON and F. D'ALCHÉ BUC. 2021, «A pseudo-metric between probability distributions based on depth-trimmed regions», *arXiv e-prints*, p. arXiv–2103. 222

TENEY, D., M. PEYRARD and E. ABBASNEJAD. 2022, «Predicting is not understanding: Recognizing and addressing underspecification in machine learning», in *European Conference on Computer Vision*, Springer, p. 458–476. 225

TOUVRON, H., M. CORD, A. EL-NOUBY, J. VERBEEK and H. J'EGOU. 2022, «Three things everyone should know about vision transformers», . 221

TOUVRON, H., M. CORD, A. SABLAYROLLES, G. SYNNAEVE and H. JÉGOU. 2021, «Going deeper with image transformers», . 221

TU, Z., H. TALEBI, H. ZHANG, F. YANG, P. MILANFAR, A. C. BOVIK and Y. LI. 2022, «Maxvit: Multi-axis vision transformer», . 221

XIE, C., J. WANG, Z. ZHANG, Y. ZHOU, L. XIE and A. YUILLE. 2017, «Adversarial examples for semantic segmentation and object detection», in *Proceedings of the IEEE international conference on computer vision*, p. 1369–1378. 222

XU, W., D. EVANS and Y. QI. 2018, «Feature squeezing: Detecting adversarial examples in deep neural networks», in *25th Annual Network and Distributed System Security Symposium*, The Internet Society. 224

YANG, R., H. MA, J. WU, Y. TANG, X. XIAO, M. ZHENG and X. LI. 2022, «Scalablevit: Rethinking the context-oriented generalization of vision transformer», . 221

YANG, Z., Z. DAI, Y. YANG, J. CARBONELL, R. R. SALAKHUTDINOV and Q. V. LE. 2019, «Xlnet: Generalized autoregressive pretraining for language understanding», *Advances in neural information processing systems*, vol. 32. 222

YI, Z., J. YU, Y. TAN and Q. WU. 2021, «A multimodal adversarial attack framework based on local and random search algorithms», *International Journal of Computational Intelligence Systems*. 221

ZHANG, H., Y. YU, J. JIAO, E. P. XING, L. E. GHAOUI and M. I. JORDAN. 2019, «Theoretically principled trade-off between robustness and accuracy», in *International Conference on Machine Learning*, p. 1–11. 219

ZHANG, J., Q. YI and J. SANG. 2022, «Towards adversarial attack on vision-language pre-training models», *arXiv preprint arXiv:2206.09391*. 221

ZHENG, Z. and P. HONG. 2018, «Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks», in *Advances in Neural Information Processing Systems 31*, p. 7924–7933. 224

Appendix A

Appendix of Chapter 3

Contents

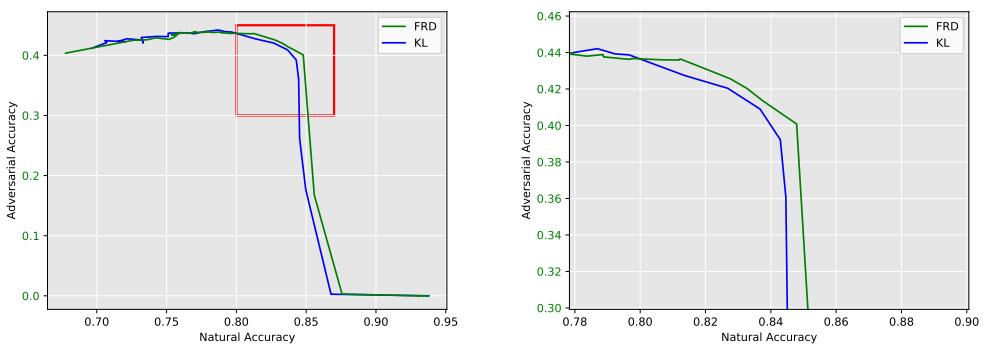
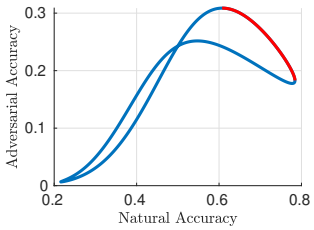
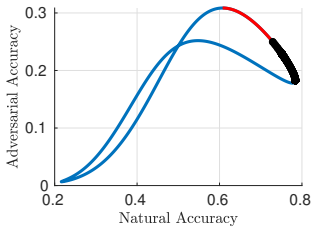


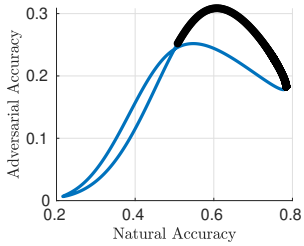
Figure A.1: Plots of all the possible points $(1-\mathrm{P}_e(\boldsymbol{\theta}), 1-\mathrm{P}'_e(\boldsymbol{\theta}))$ for ResNet-18 model on CIFAR-10. ©2022 IEEE.



(a) Pareto-optimal points. ©2022 IEEE.



(b) Hellinger



(c) FIRE

Figure A.2: Plots of all the possible points $(1-\mathrm{P}_e(\boldsymbol{\theta}), 1-\mathrm{P}'_e(\boldsymbol{\theta}))$ for the Gaussian model with $\varepsilon = 0.1$, $\boldsymbol{\mu} = [-0.0218; 0.0425]$ and $\boldsymbol{\Sigma} = [0.0212, 0.0036; 0.0036, 0.0042]$ shown in blue. In red, we show the Pareto-optimal points (Figure A.2a). In black, we show the solutions obtained by minimizing the risk $\mathrm{L}_{\mathrm{Hel}}(\boldsymbol{\theta})$ in Equation A.3 (Figure A.2b), the risk $\mathrm{L}_{\mathrm{FIRE}}(\boldsymbol{\theta})$ in Equation 3.7 (Figure A.2c).©2022 IEEE.

A.1 Comparison between the Fisher-Rao distance and the KL divergence on real data

In order to confirm on real data the difference between the Kullback-Leibler divergence and the Fisher-Rao distance, we reproduce the simulation presented on Figure 3.6 based on the CIFAR-10 dataset. We trained multiple classifier using both TRADES and FIRE methods, varying $\lambda \in [0, 50]$. We use the dataset CIFAR-10 with a ResNet-18 for the model, and train the models for 100 epochs. The optimizer is the Stochastic Gradient Descent (SGD) with a learning rate of 0.01, with a decay of 0.1 at epoch 75, 90 and 100. We also use a weight decay of 5.10^{-4}, a momentum of 0.9. The results are averaged over 2 tries.

We present the results on Figure A.1. On the left figure, we observe the entire curve, and on the right we zoomed on the zone of interest. We can see that Fisher-Rao distance presents a better trade-off between natural and adversairal accuracies than the Kullback-Leibler divergence on real data, confirming the improvements presented on Figure 3.6. For natural accuracies below 80%, the Fisher-Rao distance and the Kullback-Leibler divergence seem to behave quite similarly. For natural accuracies above 80%, which is actually the zone of interest, the improvement caused by the use of the Fisher-Rao distance seem to be quite consistent among all the training points. At fixed adversarial accuracies, the Fisher-Rao distance can increase the results by up to 1% of natural accuracies. Note that, in this simulation the Kullback-Leibler divergence can achieve a better adversarial accuracy that the Fisher-Rao distance (44.21% compared to 43.57%), but with a cost of sighly more than 2.5% for natural accuracies (78.69% compared to 81.23%). The Fisher-Rao divergence therefore achieves a better trade-off between natural and adversarial accuracies than the Kullback-Leibler divergence.

Table A.1: Comparison between Hellinger and Fisher-Rao based regularizer under white-box l_∞ threat model.

Defense	Dataset	ε	Structure	Natural	AutoAttack	Avg. Acc.	RunTime
Hellinger	MNIST	0.3	CNN	**99.31 ± 0.03**	94.03 ± 0.24	96.67 ± 0.07	2h06
FIRE			CNN	99.22 ± 0.02	**94.44 ± 0.14**	**96.83** ± 0.10	**2h06**
Hellinger	CIFAR-10	8/255	WRN-34-10	85.96 ± 0.28	50.52 ± 0.31	68.24 ± 0.15	11h02
FIRE			WRN-34-10	**85.98 ± 0.09**	**51.45 ± 0.32**	**68.72 ± 0.22**	**11h00**
Hellinger	CIFAR-100	8/255	WRN-34-10	60.79 ± 0.88	25.58 ± 0.27	43.18 ± 0.54	11h12
FIRE			WRN-34-10	**61.03 ± 0.21**	**26.42 ± 0.21**	**43.73 ± 0.12**	**11h10**

A.2 Comparison between Fisher-Rao and Hellinger distances

A.2.1 Theoretical relation

The *Hellinger* distance between two distributions q and q' on $\mathscr{Y}$ is defined as follows:

$$\mathrm{H}(q,q') \doteq \sqrt{2}\left(1-\sum_{y\in\mathscr{Y}}\sqrt{q(y)q(y)'}\right)^{1/2}. \tag{A.1}$$

Using Equation 3.15, we readily obtain the following relation between the Hellinger distance and the Fisher-Rao distance.

Theorem 4 (Relation between FRD and Hellinger distance)**.** *The* FRD *between soft-predictions $q_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x})$ and $q'_{\boldsymbol{\theta}} = q_{\boldsymbol{\theta}}(\cdot|\mathbf{x}')$, given by Equation 3.15 is related to the Hellinger distance through the relation*

$$H^2(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) = 2\left[1-\cos\left(\frac{d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}})}{2}\right)\right]. \tag{A.2}$$

Since $0 \leq d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) \leq \pi$, it is clear that $H^2(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}})$ is a monotonically increasing function of $d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}})$.

We conclude from this result that the FRD and the Hellinger distance are *theoretically equivalent* regularization mechanisms. However, it is clear that the empirical optimization of these distances may be different. This is further explored and confirmed in the following.

Experimental comparison

Since the Hellinger distance and the Fisher-Rao distance are theoretically equivalent metrics (see Subsection A.2.1), we investigate the empirical difference between those two distances. First, we perform the same simulations as those presented in Figure 3.6 but using the Hellinger distance as a robust regulatizer. Then, we perform comparison between Hellinger and FRD on real datasets (similar to the simulations given in Section 3.5.2).

Accuracy-Robustness trade-offs in the Gaussian case

As we defined the FIRE risk, it is possible to define the Hellinger risk as :

$$\mathrm{L}_{\mathrm{HEL}}(\boldsymbol{\theta}) \doteq \mathbb{E}_{p(\mathbf{x},y)}\Big[\max_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|_p\leq\varepsilon} -\log q_{\boldsymbol{\theta}}(y|\mathbf{x}) + \lambda\, \mathrm{H}^2(q_\theta(\cdot|\mathbf{x}), q_\theta(\cdot|\mathbf{x}'))\Big], \tag{A.3}$$

where $\mathrm{H}(q_\theta, q'_\theta)$ is defined in Equation A.1. In Figure 3.6b, we present the solution of the (local) Empirical Risk Minimization (ERM) for the Hellinger risk function as defined in Equation A.3 for different values of λ. As can be observed, the curve obtained for all pairs of $(1-\mathrm{P}_e(\boldsymbol{\theta}), 1-\mathrm{P}'_e(\boldsymbol{\theta}))$ covers about half of the Pareto-optimal points while the curve of all solutions for $(1-\mathrm{P}_e(\boldsymbol{\theta}), 1-\mathrm{P}'_e(\boldsymbol{\theta}))$ corresponding to FIRE risk (see Equation 3.7) covers all the Pareto-optimal points. This simulation shows that even if the Hellinger distance and the Fisher-Rao distance are theoretically equivalent, their dynamics in training are actually quite different. In addition, FRD seems to be better suited for training in the Gaussian case.

A.2.2 Comparison based on real data

We now study the empirical difference between those two distances on real datasets. We therefore perform the same simulations as those provided in Section 3.5.2 using the squared Hellinger distance between the natural and the adversarial predictions as the robustness regularizer. The results are summarized in Table A.1. Even though using the Hellinger distance as a robust regularizer performs better than using the Kullback-Leibler divergence, it still performs worse that the Fisher-Rao distance. In the case of real data, FRD appears to be better suited for training than the Hellinger distance.

In conclusion, FRD provides a better regularization objective than the Hellinger distance.

A.3 Proof of Proposition 1 and Theorem 3

A.3.1 Proof of Proposition 1

Notice that the FRD in Equation 3.9 can be written as follows:

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) = 2\Big|f(h_{\boldsymbol{\theta}}(\mathbf{x}')) - f(h_{\boldsymbol{\theta}}(\mathbf{x}))\Big|, \tag{A.4}$$

where we have defined the function $f : \mathbb{R} \to \mathbb{R}$ as $f(z) \doteq \arctan(e^{z/2})$. Notice that f is a smooth function. Using the mean value theorem, we have

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) = 2|f'(c)| \cdot \left|h_{\boldsymbol{\theta}}(\mathbf{x}') - h_{\boldsymbol{\theta}}(\mathbf{x})\right|, \tag{A.5}$$

where $f'(z) = e^{z/2}/(2e^z + 2)$ is the derivative of f and c is a point in the (open) interval with endpoints $h_{\boldsymbol{\theta}}(\mathbf{x})$ and $h_{\boldsymbol{\theta}}(\mathbf{x}')$. Notice that $0 \leq f'(z) \leq 1/4$ for any z. Then, we have

$$d_{\mathrm{R},\mathscr{C}}(q_{\boldsymbol{\theta}}, q'_{\boldsymbol{\theta}}) \leq \sup_c 2|f'(c)| \cdot \left|h_{\boldsymbol{\theta}}(\mathbf{x}') - h_{\boldsymbol{\theta}}(\mathbf{x})\right| \tag{A.6}$$

$$= \frac{1}{2}\left|h_{\boldsymbol{\theta}}(\mathbf{x}') - h_{\boldsymbol{\theta}}(\mathbf{x})\right|. \tag{A.7}$$

A.3.2 Proof of Theorem 3

Consider first the *Hellinger* distance between two distributions q and q' over $\mathscr{Y}$, defined as follows:

$$\mathrm{H}(q, q') \doteq \sqrt{2}\left(1 - \sum_{y \in \mathscr{Y}} \sqrt{q(y)q(y)'}\right)^{1/2}.$$

Using Equation 3.15, we readily obtain the following relation:

$$\mathrm{H}(q, q') = \sqrt{2}\left[1 - \cos\left(\frac{d_{\mathrm{R},\mathscr{C}}(q, q')}{2}\right)\right]^{1/2}. \tag{A.8}$$

We now use the following inequality relating the Hellinger distance and the KL divergence [TSYBAKOV, 2008, Lemma 2.4]:

$$\mathrm{H}^2(q, q') \leq \mathrm{KL}(q, q'). \tag{A.9}$$

Using the relation Equation A.8, we obtain the desired bound in Equation 3.16. The second-order approximation (cf. Equation 3.17) follows directly from CALIN and UDRIŞTE [2014][Theorem 4.4.5].

A.4 References

CALIN, O. and C. UDRIŞTE. 2014, *Geometric modeling in probability and statistics*, Springer. 237

TSYBAKOV, A. B. 2008, *Introduction to nonparametric estimation*, Springer Science & Business Media. 237

Appendix B

Appendix of Chapter 6

Contents

B.1 Approximation algorithms

In this part, we present algorithms, originally proposed in CHEN and collab. [2015] and adapted to our problem, that are used in steps 3 and 4 in HAMPER (see Algorithm Algorithm 5 for the training and Algorithm Algorithm 6 for the testing). Algorithm 7 shows the different steps to compute the scores for our HAMPER method

Algorithm 5 Training algorithm for the approximation of D_{HM}.

INPUT: : sample $\widetilde{\mathscr{S}}_c^{\ell} = \{\mathbf{z}_{\ell,i} \in \widetilde{\mathscr{S}}^{\ell} : y_i = c\}$.

INPUT: : Number of halfspaces K; sub-sample size n_s; hyperparameter λ.

for $k = 1, \ldots, \mathrm{K}$ **do**

  Draw $\widetilde{\mathscr{S}}_{c,n_s}^{\ell}$, a sub-sample of $\widetilde{\mathscr{S}}_c^{\ell}$ with size n_s without replacement.

  Draw randomly and uniformly a direction $\mathbf{u}_k$ in $\mathbb{S}^{d-1}$.

  Compute $\langle \mathbf{u}_k, \mathbf{z}_{\ell,i} \rangle$ for every $\mathbf{z}_{\ell,i} \in \widetilde{\mathscr{S}}_{c,n_s}^{\ell}$ such that $p_{k,i} \triangleq \langle \mathbf{u}_k, \mathbf{z}_{\ell,i} \rangle$.

  Set $\mathrm{mid}_k = \big(\max_i p_{k,i} + \min_i p_{k,i}\big)/2$ and $\mathrm{range}_k = \max_i p_{k,i} - \min_i p_{k,i}$.

  Randomly and uniformly select κ_k in $\big[\mathrm{mid}_k - \frac{\lambda}{2}\mathrm{range}_k,\ \mathrm{mid}_k + \frac{\lambda}{2}\mathrm{range}_k\big]$.

  Set $m_k^{\mathrm{left}} = \dfrac{|\{\mathbf{z}_{\ell,i} \in \widetilde{\mathscr{S}}_{c,n_s}^{\ell} :\ p_{k,i} < \kappa_k\}|}{n_s}$ and $m_k^{\mathrm{right}} = \dfrac{|\{\mathbf{z}_{\ell,i} \in \widetilde{\mathscr{S}}_{c,n_s}^{\ell} :\ p_{k,i} \geq \kappa_k\}|}{n_s}$.

end for

OUTPUT: : $\{\mathbf{u}_k, \kappa_k, m_k^{\mathrm{left}}, m_k^{\mathrm{right}}\}_{k=1}^{\mathrm{K}}$.

Algorithm 6 Testing algorithm for the approximation of D_{HM}.

INPUT: : test observation $\mathbf{z}_{\ell}$; $\{\mathbf{u}_k, \kappa_k, m_k^{\mathrm{left}}, m_k^{\mathrm{right}}\}_{k=1}^{\mathrm{K}}$.

INPUT: : HM=0.

for $k = 1, \ldots, \mathrm{K}$ **do**

  Project $\mathbf{z}_{\ell}$ onto $\mathbf{u}_k$ and such that $p_k^{\ell} = \langle \mathbf{z}_{\ell}, \mathbf{u}_k \rangle$.

  $\mathrm{HM} = \mathrm{HM} + m_k^{\mathrm{left}}\, \mathbb{1}\{p_k^{\ell} < \kappa_k\} + m_k^{\mathrm{right}}\, \mathbb{1}\{p_k^{\ell} \geq \kappa_k\}$.

end for

OUTPUT: : $\mathrm{D}_{\mathrm{HM}}(\mathbf{z}_{\ell}, \widetilde{\mathscr{S}}_c^{\ell}) = \mathrm{HM}/\mathrm{K}$.

Algorithm 7 `HAMPER`

INPUT: : test representations $\{\mathbf{z}_{\ell}\}_{\ell=1}^{\mathrm{L}-1}$; sample representations $\widetilde{\mathscr{S}}_c^{\ell} = \{\mathbf{z}_{\ell,i} \in \widetilde{\mathscr{S}}^{\ell} : y_i = c\}$.

INPUT: Number of closed halfspaces K; sub-sample size n_s; hyperparameter λ.

for $\ell = 1, \ldots, \mathrm{L}-1$ **do**

  for $c = 1, \ldots, \mathrm{C}$ **do**

    Draw K closed halfspaces containing $\mathbf{z}_{\ell}$ using Algorithm 5.

    Compute $\mathrm{D}_{\mathrm{HM}}(\mathbf{z}_{\ell}, \widetilde{\mathscr{S}}_c^{\ell})$ using Algorithm 6.

  end for

end for

Perform a linear regression to find weights $\alpha_{\ell,c}$ such that $s(\mathbf{x}) = \sum_{\ell=1}^{\mathrm{L}-1} \sum_{c=1}^{\mathrm{C}} \alpha_{\ell,c}\, \mathrm{D}_{\mathrm{HM}}(\mathbf{z}_{\ell}, \widetilde{\mathscr{S}}_c^{\ell})$.

OUTPUT: the score function s.

B.2 Formal description of essential properties of Data Depths

Formally, a data depth function is defined as follows:

$$\begin{aligned} \mathrm{D}: \quad \mathbb{R}^d \times \mathscr{P}(\mathbb{R}^d) &\longrightarrow [0,1]\,, \\ (\mathbf{x}, \mathrm{P}) &\longmapsto \mathrm{D}(\mathbf{x}, \mathrm{P}), \end{aligned} \tag{B.1}$$

where $\mathscr{P}(\mathbb{R}^d)$ denotes the space of all probability distributions on $\mathbb{R}^d$. The higher $\mathrm{D}(\mathbf{x}, \mathrm{P})$, the deeper $\mathbf{x}$ is in P. The depth-induced median of P is then defined by the set attaining $\sup_{\mathbf{x}\in\mathbb{R}^d} \mathrm{D}(\mathbf{x}, \mathrm{P})$ in the case where it exists. Since data depth naturally and in a nonparametric way defines a pre-order on $\mathbb{R}^d$ w.r.t. a probability distribution, it can be seen as a centrality-based alternative to the cumulative distribution function for multivariate data. Clearly, Equation B.1 opens the door to a variety of existing definitions Chen and collab. [2015]; Cuevas and collab. [2007]; Koshevoy and Mosler [1997]; Liu [1990]; Ramsay and collab. [2019]; Staerman and collab. [2021]; Zuo [2003]. While these differ in theoretical and practically related properties such as robustness or computational complexity, several postulates have been developed throughout the recent decades the "good" depth function should satisfy. Such properties have been thoroughly investigated in Liu [1990]; Zuo and Serfling [2000] and Dyckerhoff [2004] with slightly different sets of axioms (or postulates) to be satisfied by a depth function. They are recalled below.

($\mathbf{D}_1$) (Affine invariance) Denoting by $\mathrm{P}_\mathbf{X}$ the distribution of any r.v. $\mathbf{X}$ taking its values in $\mathbb{R}^d$, we have:

$$\forall \mathbf{x} \in \mathbb{R}^d,\ \mathrm{D}(\mathbf{A}\mathbf{x}+\mathbf{b}, \mathrm{P}_{\mathbf{A}\mathbf{X}+\mathbf{b}}) = \mathrm{D}(\mathbf{x}, \mathrm{P}_\mathbf{X}),$$

for any $d \times d$ nonsingular matrix $\mathbf{A}$ with real entries and any vector $\mathbf{b}$ in $\mathbb{R}^d$.

($\mathbf{D}_2$) (Maximality at center) For any $\mathrm{P} \in \mathscr{P}(\mathbb{R}^d)$ that has a symmetry center $\mathbf{x}^*$ (in a sense to be specified), the depth function $\mathrm{D}(., \mathrm{P})$ takes its maximum value at it:

$$\mathrm{D}(\mathbf{x}^*, \mathrm{P}) = \sup_{\mathbf{x}\in\mathbb{R}^d} \mathrm{D}(\mathbf{x}, \mathrm{P}).$$

($\mathbf{D}_3$) (Monotonicity relative to deepest point) For any $\mathrm{P} \in \mathscr{P}(\mathbb{R}^d)$ with deepest point $\mathbf{x}^*$, the depth at any point $\mathbf{x}$ in $\mathbb{R}^d$ decreases as one moves away from $\mathbf{x}^*$ along any ray passing through it:

$$\forall \xi \in [0,1],\ \mathrm{D}(\mathbf{x}^*, \mathrm{P}) \geq \mathrm{D}(\mathbf{x}^* + \xi(\mathbf{x} - \mathbf{x}_\mathrm{P}), \mathrm{P}).$$

($\mathbf{D}_4$) (VANISHING AT INFINITY) For any $P \in \mathscr{P}(\mathbb{R}^d)$, the depth function D vanishes at infinity:

$$D(\mathbf{x}, P) \to 0, \text{ as } ||\mathbf{x}|| \to \infty.$$

These properties introduced in ZUO and SERFLING [2000] lead to the following definition of a data depth function.

Definition B.2.1. A function $D : \mathbb{R}^d \times \mathscr{P}(\mathbb{R}^d) \longrightarrow [0,1]$ is a statistical depth function if it satisfies ($\mathbf{D}_1 - \mathbf{D}_4$).

Discussion on properties. The affine invariance property includes common transformations such as orthogonal, translation or scaling, and is useful in applications providing independence w.r.t. measurement units and coordinate system. For distributions having a uniquely defined center (e.g. symmetry center $\mathbf{x}^*$), data depths should be maximized at this center, as stated by ($\mathbf{D}_2$). The property ($\mathbf{D}_3$) is a consequence of the center-outward ordering construction of data depth. When a point $\mathbf{x} \in \mathbb{R}^d$ moves away from the set of elements that reach the maximum value of the depth function (potentially reduced to a single element, e.g. for symmetric distributions defined above), $D(\mathbf{x}, P)$ should decrease monotonically.

B.3 Detailed Results on Perfect Knowledge about the attacker

Table B.1: Performances on the three considered datasets SVHN, CIFAR10, and CIFAR100- of the $\texttt{HAMPER}_{\texttt{AA}}$ detector together with the results of the state-of-the-art detection methods: LID, and KD−BU, on multiple threat scenarios with multiple maximal perturbations ε. The best results among the detectors are shown in **bold**. The results are presented as: AUROC↑ $_{\pm\text{FPR}\downarrow_{95\%}\,\%}$. * stipulates the non-gradient based attacks.

Norm L_1	LID			KD−BU			$\texttt{HAMPER}_{\texttt{AA}}$		
	SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100
$\underline{\text{PGD}_1}$									
ε = 5 (ε⋆ = 40)	78.6 ±76.9	63.5 ±84.3	49.6 ±94.1	81.6 ±57.8	32.2 ±98.4	40.5 ±96.3	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 10 (ε⋆ = 500)	66.6 ±89.1	52.2 ±92.9	36.0 ±97.1	75.0 ±70.0	41.1 ±96.9	40.2 ±98.6	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 15 (ε⋆ = 1000)	47.2 ±94.8	48.9 ±93.7	76.2 ±66.2	76.7 ±83.6	64.9 ±91.4	64.9 ±84.6	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 20 (ε⋆ = 1500)	61.6 ±92.7	46.8 ±95.0	90.1 ±35.0	84.3 ±64.1	77.9 ±83.7	72.3 ±80.7	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 25 (ε⋆ = 2000)	65.7 ±92.0	47.0 ±95.0	95.4 ±22.2	88.8 ±55.2	87.8 ±70.5	80.9 ±74.0	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 30 (ε⋆ = 2500)	61.6 ±94.1	68.9 ±82.5	96.4 ±17.1	91.2 ±54.6	94.2 ±32.0	88.1 ±64.0	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 40 (ε⋆ = 5000)	73.2 ±91.3	77.7 ±70.2	99.8 ±0.1	95.3 ±25.5	98.7 ±5.0	99.7 ±0.2	**100** ±0.0	**100** ±0.0	**100** ±0.0
Norm L_2	SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100
$\underline{\text{PGD}_2}$									
ε = 0.125 (ε⋆ = 5)	80.8 ±76.0	63.6 ±84.3	45.3 ±95.2	84.8 ±48.5	30.3 ±98.6	40.9 ±96.0	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.25 (ε⋆ = 10)	74.8 ±79.1	59.0 ±86.3	39.7 ±96.5	76.6 ±66.6	40.6 ±98.6	41.3 ±96.2	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.3125 (ε⋆ = 15)	68.9 ±84.3	51.2 ±93.2	35.5 ±97.0	76.9 ±68.3	39.2 ±97.8	40.9 ±97.8	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.5 (ε⋆ = 20)	64.3 ±89.3	47.7 ±95.1	76.6 ±66.1	80.2 ±83.5	78.2 ±83.1	39.4 ±99.9	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 1 (ε⋆ = 30)	79.1 ±79.6	69.3 ±87.2	83.8 ±50.0	95.4 ±28.9	98.8 ±3.6	65.9 ±84.2	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 1.5 (ε⋆ = 40)	84.7 ±70.7	89.2 ±45.9	92.5 ±27.8	98.2 ±5.7	99.9 ±0.0	73.1 ±79.2	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 2 (ε⋆ = 50)	87.3 ±72.0	95.0 ±23.9	94.9 ±23.3	99.2 ±0.0	99.9 ±0.0	82.6 ±72.2	**100** ±0.0	**100** ±0.0	**100** ±0.0
$\underline{\text{DeepFool}^*}$									
No ε	93.2 ±29.4	71.4 ±81.1	96.2 ±12.2	95.0 ±15.9	85.7 ±73.5	70.9 ±72.5	**100** ±0.0	**100** ±0.0	**100** ±0.0
$\underline{\text{CW}_2{}^*}$									
ε = 0.01	61.9 ±92.8	51.0 ±94.0	31.7 ±98.9	42.7 ±88.8	45.5 ±93.4	35.9 ±96.8	**100** ±0.0	**100** ±0.0	**100** ±0.0
$\underline{\text{HOP}}$									
ε = 0.1	87.8 ±64.3	69.3 ±82.3	68.7 ±88.2	94.3 ±16.8	87.1 ±73.5	71.2 ±81.7	**100** ±0.0	**100** ±0.0	**100** ±0.0
Norm L_∞	SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100
$\underline{\text{PGD}_\infty}$									
ε = 0.03125	93.7 ±25.5	86.1 ±47.1	47.4 ±95.2	95.6 ±27.3	99.0 ±3.5	58.6 ±88.1	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.0625	98.7 ±3.7	94.6 ±25.2	48.8 ±93.9	99.6 ±0.0	**100** ±0.0	61.3 ±86.3	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.125	99.7 ±0.8	97.7 ±11.2	50.3 ±92.5	**100** ±0.0	**100** ±0.0	68.3 ±83.5	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.25	99.6 ±2.0	99.0 ±3.7	74.5 ±77.0	**100** ±0.0	**100** ±0.0	79.4 ±76.1	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.3125	99.4 ±3.2	99.1 ±3.5	77.9 ±75.6	**100** ±0.0	**100** ±0.0	83.3 ±72.5	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.5	98.7 ±6.2	99.6 ±1.1	87.2 ±56.3	**100** ±0.0	**100** ±0.0	90.8 ±56.1	**100** ±0.0	**100** ±0.0	**100** ±0.0
$\underline{\text{BIM}}$									
ε = 0.03125	91.5 ±32.6	79.7 ±63.3	47.4 ±95.2	92.2 ±40.4	95.8 ±22.0	58.7 ±88.4	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.0625	98.5 ±6.3	89.2 ±42.8	48.6 ±94.0	99.2 ±0.6	**100** ±0.0	60.7 ±86.8	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.125	99.6 ±1.6	95.8 ±21.4	49.9 ±92.8	99.9 ±0.0	**100** ±0.0	67.1 ±83.6	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.25	99.7 ±1.2	98.6 ±4.7	74.3 ±78.3	**100** ±0.0	**100** ±0.0	78.8 ±76.5	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.3125	99.4 ±3.2	99.1 ±3.5	78.3 ±73.8	**100** ±0.0	**100** ±0.0	83.4 ±72.8	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.5	99.2 ±4.3	99.7 ±1.0	85.4 ±64.1	**100** ±0.0	**100** ±0.0	91.6 ±53.6	**100** ±0.0	**100** ±0.0	**100** ±0.0
$\underline{\text{FGSM}}$									
ε = 0.03125	95.4 ±19.4	86.6 ±49.0	48.9 ±94.4	87.8 ±44.0	24.9 ±99.7	40.0 ±96.5	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.0625	99.2 ±3.8	97.3 ±12.7	47.3 ±94.2	90.7 ±31.3	81.4 ±75.4	38.5 ±96.9	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.125	99.7 ±0.2	99.4 ±2.9	47.2 ±92.0	92.7 ±22.6	93.0 ±46.3	36.0 ±97.9	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.25	99.8 ±0.0	98.4 ±3.5	82.3 ±64.2	93.6 ±18.0	98.8 ±5.0	67.1 ±80.6	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.3125	99.4 ±0.0	99.1 ±1.8	86.7 ±54.3	6.2 ±99.5	99.2 ±3.2	69.3 ±77.9	**100** ±0.0	**100** ±0.0	**100** ±0.0
ε = 0.5	99.9 ±0.0	**100** ±0.0	93.7 ±30.2	5.8 ±99.5	99.6 ±1.6	73.3 ±73.9	**100** ±0.0	**100** ±0.0	**100** ±0.0
$\underline{\text{CW}_\infty{}^*}$									
ε = 0.3125	85.9 ±51.3	71.5 ±80.5	78.1 ±81.3	90.0 ±32.7	79.5 ±76.0	67.7 ±79.7	**100** ±0.0	**100** ±0.0	**100** ±0.0
$\underline{\text{SA}}$									
ε= 0.125	92.3 ±31.8	93.1 ±45.6	84.6 ±82.0	93.0 ±22.5	90.0 ±73.4	68.9 ±78.4	**100** ±0.0	**100** ±0.0	**100** ±0.0
No norm	SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100
No ε	99.1 ±4.4	91.7 ±36.6	98.4 ±4.2	92.8 ±21.5	81.4 ±76.2	76.1 ±61.3	**100** ±0.0	**100** ±0.0	**100** ±0.0

B.4 Detailed Results on No Knowledge about the attacker

Table B.2: Performances on the three considered datasets - SVHN, CIFAR10, and CIFAR100 - of the $\texttt{HAMPER}_{\texttt{BA}}$ detector together with the results of the state-of-the-art detection method: NSS, on multiple threat scenarios with multiple maximal perturbations ε. The best results among the detectors are shown in **bold**. The results are presented as: AUROC↑ $_{\pm\text{FPR}\downarrow_{95\%}\,\%}$. * stipulates the non-gradient based attacks.

Norm L^1	NSS			$\texttt{HAMPER}_{\texttt{BA}}$		
	SVHN	CIFAR10	CIFAR100⋆	SVHN	CIFAR10	CIFAR100⋆
PGD_1						
ε = 5 (ε⋆ = 40)	48.6 $_{\pm 95.1}$	50.1 $_{\pm 94.3}$	51.6 $_{\pm 94.3}$	**95.7** $_{\pm 20.4}$	**88.4** $_{\pm 47.5}$	**99.9** $_{\pm 0.4}$
ε = 10 (ε⋆ = 500)	51.5 $_{\pm 94.6}$	56.7 $_{\pm 90.0}$	53.1 $_{\pm 94.2}$	**87.6** $_{\pm 51.0}$	**90.4** $_{\pm 38.8}$	**99.9** $_{\pm 0.3}$
ε = 15 (ε⋆ = 1000)	59.4 $_{\pm 91.7}$	62.5 $_{\pm 85.2}$	58.3 $_{\pm 93.3}$	**95.1** $_{\pm 23.5}$	**89.2** $_{\pm 49.0}$	**99.9** $_{\pm 0.2}$
ε = 20 (ε⋆ = 1500)	69.2 $_{\pm 85.1}$	67.7 $_{\pm 80.3}$	66.5 $_{\pm 91.8}$	**96.0** $_{\pm 21.8}$	**98.3** $_{\pm 8.5}$	**99.9** $_{\pm 0.2}$
ε = 25 (ε⋆ = 2000)	78.1 $_{\pm 71.7}$	72.0 $_{\pm 74.5}$	75.5 $_{\pm 89.2}$	**96.4** $_{\pm 19.8}$	**98.3** $_{\pm 8.2}$	**99.9** $_{\pm 0.2}$
ε = 30 (ε⋆ = 2500)	84.8 $_{\pm 55.5}$	75.9 $_{\pm 68.6}$	83.7 $_{\pm 84.4}$	**95.7** $_{\pm 27.5}$	**98.8** $_{\pm 5.4}$	**100** $_{\pm 0.1}$
ε = 40 (ε⋆ = 5000)	92.9 $_{\pm 24.3}$	82.1 $_{\pm 57.1}$	99.4 $_{\pm 0.2}$	**97.9** $_{\pm 10.1}$	**99.3** $_{\pm 2.5}$	**100** $_{\pm 0.0}$
Norm L_2	SVHN	CIFAR10	CIFAR100⋆	SVHN	CIFAR10	CIFAR100⋆
PGD_2						
ε = 0.125 (ε⋆ = 5)	49.2 $_{\pm 95.0}$	49.7 $_{\pm 94.2}$	51.5 $_{\pm 94.3}$	**93.0** $_{\pm 27.7}$	**94.6** $_{\pm 26.1}$	**99.9** $_{\pm 0.6}$
ε = 0.25 (ε⋆ = 10)	49.6 $_{\pm 94.9}$	55.7 $_{\pm 90.5}$	51.7 $_{\pm 94.0}$	**90.4** $_{\pm 37.7}$	**89.6** $_{\pm 45.1}$	**99.9** $_{\pm 0.3}$
ε = 0.3125 (ε⋆ = 15)	52.5 $_{\pm 94.1}$	59.0 $_{\pm 88.2}$	52.8 $_{\pm 94.2}$	**91.9** $_{\pm 34.0}$	**79.9** $_{\pm 69.9}$	**99.8** $_{\pm 0.8}$
ε = 0.5 (ε⋆ = 20)	66.4 $_{\pm 87.4}$	67.5 $_{\pm 79.9}$	54.4 $_{\pm 94.4}$	**94.0** $_{\pm 30.7}$	**94.7** $_{\pm 30.6}$	**99.9** $_{\pm 0.2}$
ε = 1 (ε⋆ = 30)	92.1 $_{\pm 29.6}$	83.1 $_{\pm 54.4}$	59.1 $_{\pm 93.6}$	**98.8** $_{\pm 6.5}$	**99.4** $_{\pm 3.3}$	**100** $_{\pm 0.1}$
ε = 1.5 (ε⋆ = 40)	98.0 $_{\pm 5.9}$	91.7 $_{\pm 32.8}$	67.2 $_{\pm 92.4}$	**98.9** $_{\pm 4.4}$	**100** $_{\pm 0.0}$	**99.9** $_{\pm 0.2}$
ε = 2 (ε⋆ = 50)	99.4 $_{\pm 1.6}$	96.2 $_{\pm 16.1}$	77.5 $_{\pm 88.4}$	**99.5** $_{\pm 2.4}$	**100** $_{\pm 0.0}$	**100** $_{\pm 0.1}$
DeepFool*						
No ε	58.2 $_{\pm 93.4}$	55.9 $_{\pm 92.3}$	73.0 $_{\pm 72.6}$	**90.8** $_{\pm 29.0}$	**79.7** $_{\pm 62.5}$	**100** $_{\pm 0.0}$
CW_2*						
ε = 0.01	61.8 $_{\pm 92.0}$	56.0 $_{\pm 91.2}$	64.6 $_{\pm 90.0}$	**92.7** $_{\pm 30.9}$	**89.2** $_{\pm 44.6}$	**99.9** $_{\pm 0.1}$
HOP						
ε = 0.1	87.6 $_{\pm 64.0}$	65.4 $_{\pm 87.9}$	73.2 $_{\pm 87.9}$	**93.8** $_{\pm 22.5}$	**95.6** $_{\pm 19.6}$	**99.8** $_{\pm 0.5}$
Norm L_∞	SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100
PGD_∞						
ε = 0.03125	**99.3** $_{\pm 1.7}$	91.3 $_{\pm 34.2}$	53.1 $_{\pm 93.2}$	97.4 $_{\pm 13.8}$	**99.4** $_{\pm 2.3}$	**99.9** $_{\pm 0.4}$
ε = 0.0625	**99.9** $_{\pm 0.2}$	99.0 $_{\pm 4.4}$	55.9 $_{\pm 91.8}$	99.5 $_{\pm 2.0}$	**99.3** $_{\pm 3.6}$	**100** $_{\pm 0.1}$
ε = 0.125	**99.9** $_{\pm 0.2}$	**99.9** $_{\pm 0.3}$	61.5 $_{\pm 89.9}$	**99.9** $_{\pm 0.3}$	99.8 $_{\pm 0.7}$	**99.9** $_{\pm 0.4}$
ε = 0.25	**99.9** $_{\pm 0.2}$	99.9 $_{\pm 0.1}$	71.3 $_{\pm 84.9}$	99.7 $_{\pm 1.7}$	**100** $_{\pm 0.0}$	**99.9** $_{\pm 0.4}$
ε = 0.3125	99.9 $_{\pm 0.2}$	**99.9** $_{\pm 0.1}$	75.4 $_{\pm 81.3}$	**100** $_{\pm 0.1}$	99.8 $_{\pm 0.8}$	**99.8** $_{\pm 0.8}$
ε = 0.5	99.9 $_{\pm 0.2}$	99.9 $_{\pm 0.1}$	84.7 $_{\pm 67.6}$	**100** $_{\pm 0.2}$	**100** $_{\pm 0.1}$	**99.8** $_{\pm 0.5}$
BIM						
ε = 0.03125	**99.0** $_{\pm 3.0}$	89.3 $_{\pm 41.4}$	53.0 $_{\pm 93.3}$	96.7 $_{\pm 21.2}$	**96.8** $_{\pm 14.4}$	**99.9** $_{\pm 0.3}$
ε = 0.0625	**99.8** $_{\pm 0.3}$	97.9 $_{\pm 9.2}$	55.8 $_{\pm 91.9}$	99.2 $_{\pm 3.8}$	**99.8** $_{\pm 0.8}$	**99.9** $_{\pm 0.2}$
ε = 0.125	**99.9** $_{\pm 0.2}$	99.7 $_{\pm 0.9}$	61.4 $_{\pm 90.3}$	**99.9** $_{\pm 0.7}$	**99.8** $_{\pm 1.1}$	**99.9** $_{\pm 0.4}$
ε = 0.25	99.9 $_{\pm 0.2}$	**99.9** $_{\pm 0.1}$	71.2 $_{\pm 85.1}$	**100** $_{\pm 0.0}$	**99.9** $_{\pm 0.4}$	**99.9** $_{\pm 0.4}$
ε = 0.3125	**99.9** $_{\pm 0.2}$	**99.9** $_{\pm 0.1}$	75.3 $_{\pm 81.0}$	**99.9** $_{\pm 0.6}$	99.8 $_{\pm 1.0}$	**99.9** $_{\pm 0.4}$
ε = 0.5	**99.9** $_{\pm 0.2}$	**99.9** $_{\pm 0.1}$	84.9 $_{\pm 67.7}$	**99.9** $_{\pm 0.2}$	99.7 $_{\pm 1.4}$	**99.9** $_{\pm 0.1}$
FGSM						
ε = 0.03125	**99.6** $_{\pm 0.9}$	93.6 $_{\pm 28.2}$	53.1 $_{\pm 93.6}$	94.3 $_{\pm 26.2}$	**96.9** $_{\pm 15.7}$	**99.8** $_{\pm 0.6}$
ε = 0.0625	98.7 $_{\pm 0.2}$	**99.6** $_{\pm 1.5}$	57.1 $_{\pm 90.9}$	**95.5** $_{\pm 20.3}$	94.6 $_{\pm 27.0}$	**99.9** $_{\pm 0.2}$
ε = 0.125	82.0 $_{\pm 100}$	**99.9** $_{\pm 0.1}$	67.2 $_{\pm 87.0}$	**98.6** $_{\pm 6.5}$	99.7 $_{\pm 1.1}$	**99.9** $_{\pm 0.1}$
ε = 0.25	70.5 $_{\pm 100}$	**99.9** $_{\pm 0.1}$	82.1 $_{\pm 74.0}$	**99.4** $_{\pm 3.2}$	99.9 $_{\pm 0.5}$	**99.9** $_{\pm 0.4}$
ε = 0.3125	76.1 $_{\pm 100}$	99.9 $_{\pm 0.1}$	86.9 $_{\pm 64.5}$	**98.5** $_{\pm 7.2}$	**100** $_{\pm 0.1}$	**99.9** $_{\pm 0.4}$
ε = 0.5	88.7 $_{\pm 97.6}$	99.9 $_{\pm 0.1}$	94.1 $_{\pm 36.9}$	**99.8** $_{\pm 0.7}$	**100** $_{\pm 0.0}$	**99.9** $_{\pm 0.2}$
CW_∞*						
ε = 0.3125	67.9 $_{\pm 90.9}$	64.6 $_{\pm 89.9}$	70.0 $_{\pm 85.3}$	**90.5** $_{\pm 33.0}$	**90.5** $_{\pm 40.9}$	**99.9** $_{\pm 0.1}$
SA						
ε= 0.125	91.3 $_{\pm 82.3}$	11.6 $_{\pm 99.9}$	67.4 $_{\pm 89.3}$	**99.0** $_{\pm 4.9}$	**97.9** $_{\pm 12.7}$	**99.9** $_{\pm 0.2}$
No norm	SVHN	CIFAR10	CIFAR100	SVHN	CIFAR10	CIFAR100
No ε	**99.8** $_{\pm 6.4}$	**93.8** $_{\pm 20.2}$	92.9 $_{\pm 24.7}$	98.5 $_{\pm 0.4}$	80.3 $_{\pm 57.1}$	**100** $_{\pm 0.0}$

B.5 References

CHEN, B., K. M. TING, T. WASHIO and G. HAFFARI. 2015, «Half-space mass: a maximally robust and efficient data depth method», *Machine Learning*, vol. 100, nº 2, p. 677–699. 239, 241

CUEVAS, A., M. FEBRERO and R. FRAIMAN. 2007, «Robust estimation and classification for functional data via projection-based depth notions», *Computational Statistics*, vol. 22, nº 3, p. 481–496. 241

DYCKERHOFF, R. 2004, «Data depth satisfying the projection property», *Allgemeines Statistisches Archiv*, vol. 88, nº 2, p. 163–190. 241

KOSHEVOY, G. and K. MOSLER. 1997, «Zonoid trimming for multivariate distributions», *The Annals of Statistics*, vol. 25, nº 5, p. 1998–2017. 241

LIU, R. 1990, «On a notion of data depth based on random simplices», *The Annals of Statistics*, vol. 18, p. 405–414. 241

RAMSAY, K., S. DUROCHER and A. LEBLANC. 2019, «Integrated rank-weighted depth», *Journal of Multivariate Analysis*, vol. 173, p. 51–69. 241

STAERMAN, G., P. MOZHAROVSKYI and S. CLÉMENÇON. 2021, «Affine-invariant integrated rank-weighted depth: Definition, properties and finite sample analysis», *arXiv preprint arXiv:2106.11068*. 241

ZUO, Y. 2003, «Projection-based depth functions and associated medians», *The Annals of Statistics*, vol. 31, nº 5, p. 1460–1490. 241

ZUO, Y. and R. SERFLING. 2000, «General notions of statistical depth function», *The Annals of Statistics*, vol. 28, nº 2, p. 461–482. 241, 242

Appendix C

Appendix of Chapter 7

Contents

C.1 Training Details

We compare the different detection methods on three vision datasets: CIFAR10, CIFAR100 [KRIZHEVSKY and collab.] and Tiny ImageNet [JIAO and collab., 2019] for which we use the ViT models presented in Subsection 7.4.1 to build a classifier.

We trained two different models: a ViT, and a ResNet18. The ResNet18 has been trained on 100 epochs, with a Stochastic Gradient Descent (SGD) optimizer, with a learning rate of 0.1, a momentum of 0.9, and a weight decay of 10^{-5}. We use the base model with 16 layers (85.8 million of parameters) from `https://github.com/jeonsworld/ViT-pytorch` trained on ImageNet [DENG and collab., 2009] as our ViT classifier for CIFAR10 and CIFAR100. To train it we set the batch size to 512. The learning rate of SGD [RUDER, 2016] is set to 3×10^{-2} and we use 500 warming steps with no gradient accumulation [VASWANI and collab., 2017]. For Tiny ImageNet, we used

as the underlying classifier a ViT with 16 layers, trained by HUYNH [2022] and available at https://github.com/ehuynh1106/TinyImageNet-Transformers. Note that we only use the class token to output the layer-wise input's representations.

Remark. We compare our proposed APPROVED method with FS and MagNet, recalled in Subsection 7.2.2. We train MagNet according to its original training procedure, while FS and our APPROVED, presented in Subsection 7.3.2, do not require any training.

C.2 Approximation Algorithm

In this appendix, we display the algorithm used to compute the IRW depth (see Algorithm 8).

Algorithm 8 Approximation of the IRW depth

Initialization: test sample $\mathbf{x}$, n_{proj}, $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^\top$.

1: Construct $\mathbf{U} \in \mathbb{R}^{d \times n_{\text{proj}}}$ by sampling uniformly n_{proj} vectors $\mathbf{U}_1, \dots, \mathbf{U}_{n_{\text{proj}}}$ in $\mathbb{S}^{d-1}$

2: Compute $\mathbf{M} = \mathbf{X}\mathbf{U}$ and $\mathbf{x}^\top\mathbf{U}$

3: Compute the rank value $\sigma(j)$, the rank of $\mathbf{x}^\top\mathbf{U}$ in $\mathbf{M}_{:,j}$ for every $j \leq n_{\text{proj}}$

4: Set $\mathrm{D} = \frac{1}{n_{\text{proj}}} \sum_{j=1}^{n_{\text{proj}}} \sigma(j)$

5: **Output**: $\widetilde{\mathrm{D}}^{\mathrm{MC}}_{\mathrm{IRW}}(\mathbf{x}, \mathbf{X}) = \mathrm{D}$

Complexity. The complexity of the algorithm is detailed as follows. Line 1 requires sampling n_{proj} Gaussian samples and normalizing them in order to define unit sphere directions and can be computed in $\mathrm{O}(n_{\text{proj}}d)$. Line 2 requires $\mathrm{O}(n_{\text{proj}}dn)$ to project data on the n_{proj} unit sphere Monte-Carlo directions. Line 3 requires computing the sorting operation on n_{proj} columns of the matrix M and then leads to a complexity of $\mathrm{O}(n_{\text{proj}}n)$. Line 4 requires the computation of the mean and can be done in n_{proj} operations. Finally, the total complexity of the algorithm is then in $\mathrm{O}(n_{\text{proj}}dn)$ which is linear in all of its parameters.

Remarks. Given that the algorithm is linear in all its parameters, computing the IRW depth can be scaled to any datasets. Note that the IRW data depth makes no assumption on the training distribution. In line 3 of Algorithm 8, "rank values" consists in ranking the elements of the projection of each input on $\mathbf{U}$. This is achieved by a sorting algorithm. This step allows us to define an ordering of the projected inputs, which is used to compute the final depth score.

C.3 Time and Computational Requirements

C.3.1 To generate attacks

We here present the computational requirements to generate the attacks on the transformer, along with the required time to generate them. We use the Adversarial-Robustness Toolbox (ART) [NICOLAE and collab., 2018] to generate the attacks.

Table C.1: Resources and time needed to generate different types of attack on CIFAR10

Attack	GPUs	CPUs	Time
FGSM	V100-32G	20G	0h25
BIM	V100-32G	20G	3h13
PGD	V100-32G	20G	4h30
DF	V100-32G	20G	1h54
HOP	V100-32G	20G	47h39
CW^{∞}	V100-32G	30G	2h48
SA	V100-32G	20G	5h04
STA	V100-32G	20G	1h25

C.3.2 To deploy detectors

This section presents the computational requirements, along with the time needed to deploy each of the studied detection methods on CIFAR10. For FS and MagNet, we use the codes available at https://github.com/aldahdooh/detectors_review.

Table C.2: Resources and time needed to train and test each detection method

Method	GPUs	CPUs	Training Time	Testing Time
APPROVED	V100-32G	40G	N/A	0h11
FS	V100-32G	80G	N/A	0h53
MagNet	V100-32G	180G	3h01	0h13

C.4 Success Rates of Attacks on CIFAR10

We here report the success rate per attack for all the different threat mechanisms (i.e., PGD^1, PGD^2, PGD^{∞}, BIM, FGSM, CW^{∞}, SA, STA, DF and HOP). In orange are the attack performances on ViT while the ones on ResNet are in green (see Section 7.4 for a detailed analysis).

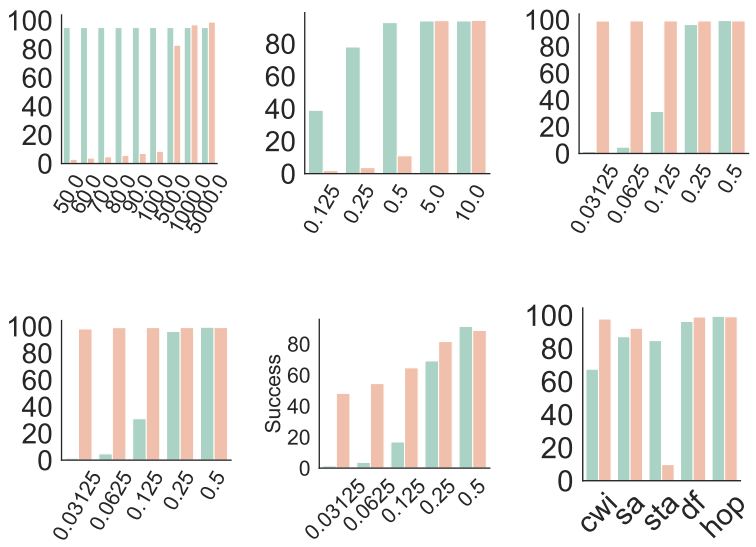


Figure C.1: Percentage of successful attacks depending on the L_p-norm constraint, the maximal perturbation ε and the attack algorithm on ResNet18 (orange) and ViT (blue).

C.5 Detailed results for CIFAR10, CIFAR100, and Tiny ImageNet

C.5.1 Detailed Tables

In Table C.3, Table C.4 and Table C.5, we present the detailed results for CIFAR10, CIFAR100 and Tiny ImageNet under multiple threats. From Table C.3, Table C.4 and Table C.5, it is straightforward to conclude that `APPROVED` significantly outperforms FS, and MagNet.

Table C.3: AUROC↑ and FPR↓$_{90\%}$ for each considered attack mechanism, L_p-norm constraint and ε on CIFAR10 for APPROVED, FS, and MagNet. The best result for each attack is shown in **bold**.

CIFAR10						
Norm L1	APPROVED		FS		MagNet	
	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$
PGD[1]						
ε = 50	**97.2**	**5.0**	77.6	37.5	53.3	90.1
ε = 60	**97.0**	**5.7**	77.4	37.5	51.6	92.1
ε = 70	**96.4**	**6.8**	78.0	31.2	51.9	92.0
ε = 80	**95.7**	**8.6**	78.1	31.2	51.3	91.9
ε = 90	**94.8**	**11.1**	78.7	31.2	52.0	91.6
ε = 100	**93.9**	**13.9**	79.0	37.5	51.6	91.6
ε = 500	80.1	50.1	**86.8**	**25.0**	49.6	90.5
ε = 1000	**93.0**	**14.2**	83.7	37.5	49.9	90.0
ε = 5000	**98.0**	**3.6**	76.0	55.2	50.1	89.9
Norm L2	APPROVED		FS		MagNet	
	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$
PGD[2]						
ε = 0.125	**97.1**	**4.5**	75.5	37.5	50.6	92.1
ε = 0.25	**97.1**	**5.5**	77.2	37.5	52.2	91.7
ε = 0.5	**92.6**	**18.1**	79.8	31.2	50.6	91.6
ε = 5	**93.3**	**13.6**	77.0	45.9	50.0	89.8
ε = 10	**94.1**	**11.5**	76.8	52.1	50.1	89.8
HOP						
ε = 0.1	**98.3**	**3.3**	74.5	25.0	53.4	83.6
DeepFool						
No ε	**86.5**	45.4	79.7	**31.2**	50.3	89.7
Norm L_∞	APPROVED		FS		MagNet	
	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$
PGD$^\infty$						
ε = 0.03125	**96.5**	**6.4**	78.7	42.9	50.3	89.6
ε = 0.0625	**99.1**	**2.1**	73.4	64.7	51.0	88.4
ε = 0.125	**99.7**	**0.8**	71.8	68.6	52.9	85.5
ε = 0.25	**99.8**	**0.5**	70.9	70.0	54.3	83.4
ε = 0.5	**99.8**	**0.5**	70.8	70.1	54.4	83.3
BIM						
ε = 0.03125	**88.3**	**27.0**	74.0	64.5	50.3	89.6
ε = 0.0625	**97.1**	**5.4**	70.2	72.3	50.7	88.9
ε = 0.125	**99.0**	**2.2**	70.0	72.2	51.8	87.2
ε = 0.25	**99.7**	**0.7**	70.7	70.5	53.6	84.4
ε = 0.5	**99.9**	**0.2**	71.2	68.4	56.4	80.1
FGSM						
ε = 0.03125	**78.1**	69.5	75.2	**38.8**	51.9	88.1
ε = 0.0625	**82.4**	60.2	77.2	**37.5**	53.0	86.1
ε = 0.125	**93.1**	**16.6**	78.9	31.2	57.3	79.2
ε = 0.25	**99.1**	**1.6**	69.6	25.0	70.6	54.8
ε = 0.5	**99.7**	**0.6**	67.7	31.2	80.4	18.0
SA						
ε = 0.125	**98.2**	**3.3**	72.0	25.0	55.1	82.4
CW$^\infty$						
ε = 0.3125	**90.4**	**30.6**	78.8	37.5	50.6	89.3
No Norm	APPROVED		FS		MagNet	
	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$
STA						
No ε	**94.9**	**10.5**	78.8	37.5	39.4	93.5

Table C.4: AUROC↑ and FPR↓$_{90\%}$ for each considered attack mechanism, L$_p$-norm constraint and ε on CIFAR100 for APPROVED, FS and MagNet. The best result for each attack is shown in **bold**.

CIFAR100						
Norm L1	APPROVED		FS		MagNet	
	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$
PGD[1]						
ε = 50	**83.5**	**39.3**	65.5	56.2	50.5	90.5
ε = 60	**82.4**	**41.0**	66.6	56.2	50.5	90.3
ε = 70	**81.2**	**45.3**	67.4	50.0	50.0	90.4
ε = 80	**79.8**	**47.8**	68.3	50.0	50.0	90.4
ε = 90	**78.4**	**50.0**	69.2	**50.0**	50.2	90.3
ε = 100	**77.0**	54.0	70.1	**50.0**	50.1	90.4
ε = 500	58.1	75.5	**79.3**	**50.0**	50.0	90.0
ε = 1000	78.3	**44.9**	**80.0**	62.5	50.0	89.9
ε = 5000	**86.1**	**29.4**	74.0	75.0	50.0	89.8
Norm L2	APPROVED		FS		MagNet	
	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$
PGD[2]						
ε = 0.125	**84.3**	**38.3**	64.6	56.2	50.8	90.8
ε = 0.25	**82.7**	**41.4**	66.2	56.2	50.8	90.1
ε = 0.5	**73.9**	59.1	72.0	**50.0**	50.3	90.0
ε = 5	**78.6**	**43.5**	75.1	75.0	50.0	89.9
ε = 10	**79.4**	**41.0**	74.4	75.0	50.0	89.9
HOP						
ε = 0.1	**89.1**	**24.8**	62.7	50.0	52.1	84.5
DeepFool						
No ε	**75.5**	59.9	62.2	**50.0**	50.0	89.9
Norm L$_\infty$	APPROVED		FS		MagNet	
	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$
PGD$^\infty$						
ε = 0.03125	75.4	**51.5**	**76.0**	74.8	50.2	89.7
ε = 0.0625	**88.1**	**26.0**	68.9	75.0	50.6	88.9
ε = 0.125	**93.3**	**14.9**	65.5	75.0	52.1	86.5
ε = 0.25	**94.4**	**12.8**	64.3	75.0	53.0	84.9
ε = 0.5	**89.7**	**26.4**	64.2	75.0	53.1	84.8
BIM						
ε = 0.03125	63.1	**72.9**	**67.6**	75.0	50.2	89.7
ε = 0.0625	**70.5**	**64.8**	63.0	81.1	50.5	89.2
ε = 0.125	**87.2**	**28.1**	62.1	82.7	51.3	87.8
ε = 0.25	**93.2**	**15.4**	63.7	75.4	52.5	85.7
ε = 0.5	**96.5**	**8.3**	65.3	75.0	54.6	82.2
FGSM						
ε = 0.03125	**80.8**	**48.1**	61.9	62.5	51.0	88.8
ε = 0.0625	**86.5**	**33.0**	61.3	61.4	52.1	86.8
ε = 0.125	**90.4**	**24.0**	54.8	50.0	55.8	80.2
ε = 0.25	**95.7**	**10.3**	49.6	50.0	66.4	60.4
ε = 0.5	**98.6**	**4.1**	46.2	56.2	86.6	24.2
SA						
ε = 0.125	**89.6**	**26.0**	63.3	50.0	54.9	82.6
CW$^\infty$						
ε = 0.3125	**81.7**	**42.2**	67.0	50.0	50.0	89.8
No Norm	APPROVED		FS		MagNet	
	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$
STA						
No ε	**87.4**	**32.1**	65.4	50.0	38.3	92.8

Table C.5: AUROC↑ and FPR↓$_{90\%}$ for each considered attack mechanism, L_p-norm constraint and ε on Tiny ImageNet for APPROVEDand FS. The best result for each attack is shown in **bold**.

Tiny ImageNet						
Norm L1	APPROVED		FS		MagNet	
	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$
PGD[1]						
ε = 50	**74.2**	**61.1**	44.8	81.6	50.4	88.9
ε = 60	**74.3**	**60.7**	45.0	81.8	50.3	88.9
ε = 70	**74.8**	**60.7**	45.1	82.0	50.0	89.0
ε = 80	**74.7**	**60.5**	45.1	82.3	49.6	88.9
ε = 90	**74.9**	**59.8**	45.0	82.2	49.7	89.3
ε = 100	**74.6**	**59.4**	44.9	82.0	49.6	89.0
ε = 500	**76.5**	**59.7**	60.7	71.7	48.0	93.1
ε = 1000	**74.2**	**59.4**	73.7	62.4	47.6	92.0
ε = 5000	78.2	51.8	**83.2**	**50.0**	49.1	90.3
Norm L2	APPROVED		FS		MagNet	
	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$	AUROC↑	FPR↓$_{90\%}$
PGD[2]						
ε = 0.125	**74.2**	**60.2**	45.2	81.4	50.2	88.7
ε = 0.25	**75.0**	**57.2**	45.2	81.8	49.3	89.7
ε = 0.5	**75.7**	**53.4**	47.1	79.5	49.6	91.0
ε = 5	74.3	60.6	**77.9**	**57.5**	48.7	91.0
ε = 10	74.4	59.7	**78.1**	**57.7**	48.8	90.9
HOP						
ε = 0.1	**87.1**	**31.8**	59.1	76.3	52.7	83.8
Norm L_∞	APPROVED		FS		MagNet	
	AUROC	FPR	AUROC	FPR	AUROC	FPR
PGD$^\infty$						
ε = 0.03125	89.6	28.8	**96.0**	**8.2**	49.7	90.0
ε = 0.0625	**99.1**	**1.9**	93.8	11.9	49.8	89.9
ε = 0.125	**99.9**	**0.0**	89.2	47.1	49.9	89.6
ε = 0.25	**99.9**	**0.0**	85.5	73.6	50.0	89.5
ε = 0.5	**99.9**	**0.0**	83.6	82.2	50.1	89.4
BIM						
ε = 0.03125	80.7	**43.1**	**86.0**	44.8	49.5	90.1
ε = 0.0625	**95.1**	**15.1**	90.3	33.4	49.9	89.9
ε = 0.125	**99.6**	**1.0**	87.4	61.4	49.9	89.8
ε = 0.25	**99.9**	**0.0**	84.9	79.9	50.0	89.5
ε = 0.5	**99.9**	**0.0**	83.9	82.5	50.2	89.1
FGSM						
ε = 0.03125	**74.5**	**55.9**	56.3	75.5	49.7	90.2
ε = 0.0625	**80.8**	**43.5**	58.0	71.8	50.4	89.6
ε = 0.125	**87.1**	**30.4**	53.6	75.1	50.9	88.7
ε = 0.25	**91.1**	**22.3**	48.1	78.8	52.6	86.2
ε = 0.5	**94.4**	**15.2**	50.9	74.2	60.7	72.1
SA						
ε = 0.125	**77.0**	**49.1**	48.7	78.5	50.6	89.4
No Norm	APPROVED		FS		MagNet	
	AUROC	FPR	AUROC	FPR	AUROC	FPR
STA						
No ε	**80.2**	**42.5**	53.0	77.5	34.9	95.6

C.6 Per Class Analysis

As for CIFAR10 (see Section 3.5), the detector performances depend on the predicted class. Some classes are easy to detect (i.e., classes 0, 21, 53, 75, and 94), others are more difficult (i.e., classes 3, 10, 33, 47, 60, 74, and 93). Some have low variance (i.e., 0, 1, 24, 34, 75, 82 and, 94) while others have an extremely large dispersion (i.e., 11, 35,47, 52, 96, and 98).

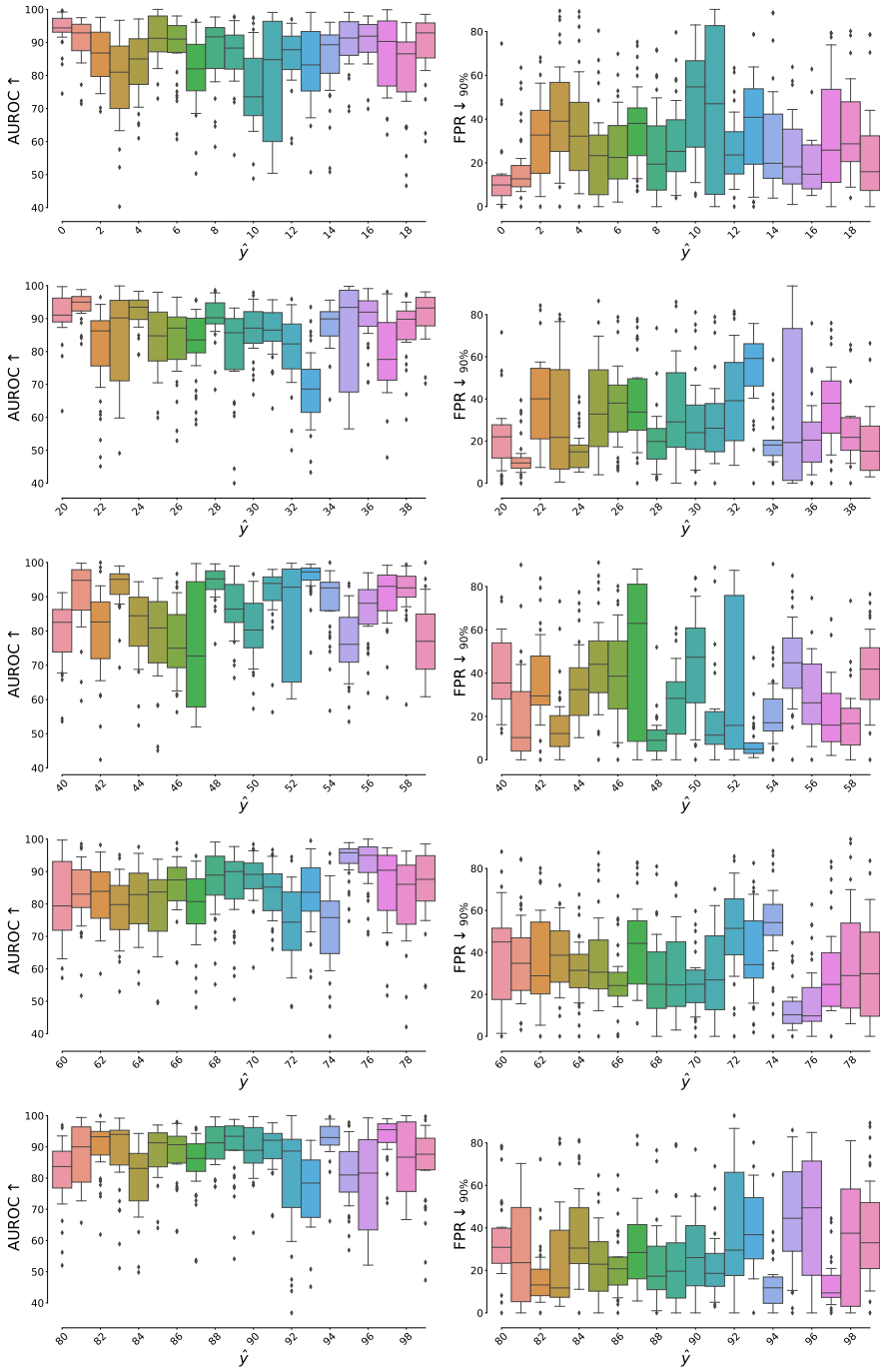


Figure C.2: APPROVED's AUROC↑ and FPR$\downarrow_{90\%}$ per class, averaged over the attacks on CIFAR100.

C.7 References

DENG, J., W. DONG, R. SOCHER, L.-J. LI, K. LI and L. FEI-FEI. 2009, «Imagenet: A large-scale hierarchical image database», in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, p. 248–255. 247

HUYNH, E. 2022, «Vision transformers in 2022: An update on tiny imagenet», *arXiv preprint arXiv:2205.10660*. 248

JIAO, X., Y. YIN, L. SHANG, X. JIANG, X. CHEN, L. LI, F. WANG and Q. LIU. 2019, «Tinybert: Distilling bert for natural language understanding», *arXiv preprint arXiv:1909.10351*. 247

KRIZHEVSKY, A., V. NAIR and G. HINTON. «Cifar-100 (canadian institute for advanced research)», . 247

NICOLAE, M.-I., M. SINN, M. N. TRAN, B. BUESSER, A. RAWAT, M. WISTUBA, V. ZANTEDESCHI, N. BARACALDO, B. CHEN, H. LUDWIG, I. MOLLOY and B. EDWARDS. 2018, «Adversarial robustness toolbox v1.2.0», *CoRR*, vol. 1807.01069. 249

RUDER, S. 2016, «An overview of gradient descent optimization algorithms», *arXiv preprint arXiv:1609.04747*. 247

VASWANI, A., N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER and I. POLOSUKHIN. 2017, «Attention is all you need», *Advances in neural information processing systems*, vol. 30. 247

Appendix D

Résumé Étendu

Au cours de la dernière décennie, l'apprentissage profond a été à l'origine de percées dans de nombreux domaines différents, tels que le traitement du langage naturel, la vision par ordinateur et la reconnaissance vocale. Cependant, il est désormais connu que les modèles basés sur l'apprentissage profond sont extrêmement sensibles aux perturbations, en particulier lorsque la perturbation est bien conçue et générée par un agent malveillant. Cette faiblesse des réseaux neuronaux profonds tend à empêcher leur utilisation dans des applications critiques, où des informations sensibles sont disponibles, ou lorsque le système interagit directement avec la vie quotidienne des gens. Dans cette thèse, nous nous concentrons sur la protection des réseaux neuronaux profonds contre les agents malveillants de deux manières principales.

La première méthode vise à protéger un modèle des attaques en augmentant sa robustesse, c'est-à-dire la capacité du modèle à prédire la bonne classe même en cas d'attaques. Nous observons que la sortie d'un réseau neuronal profond forme une variété statistique et que la décision est prise sur cette variété. Nous exploitons cette connaissance en utilisant la mesure de Fisher-Rao, qui calcule la distance géodésique entre deux distributions de probabilité sur la variété statistique auquel elles appartiennent.

Dans un premier temps, nous développons une méthode pour le cas de la classification d'images. Nous derivons la formule explicite de la mesure de Fisher-Rao dans deux cas différents: pour la classification binaire résultant d'un modèle paramétique, et pour la classification multi-classe résultant d'un modèle paramétrique. Nous présentons ensuite les connexions entre cette mesure, et des mesures plus connues, comme la divergence de Kullback-Leibler, ou la divergence d'Hellinger. Nous présentons ensuite notre méthode pour utiliser la mesure de Fisher-Rao pour régulariser la fonction coût utilisée lors de l'apprentissage et augmenter la robustesse du modèle. Finalement, expérimentalement parlant, nous montrons les avantages de l'utilisation de la mesure de Fisher-Rao comparé à l'utilisation d'autres métriques.

Dans un second temps, nous adaptons cette méthode à une autre application critique: les réseaux intelligents (Smart Grids), qui, en raison de divers besoins de la surveillance et de service, reposent sur des composants cybernétiques, tels qu'un estimateur d'état, ce qui les rend sensibles aux attaques. Nous construisons donc des estimateurs d'état robustes en utilisant des autoencodeurs variationnels et l'extension de notre méthode proposée au cas de la régression. Que ce soit dans le cas des hypothèses linéraire (appelé modèle DC) ou sans hypothèses (appelé modèle AC), nous montrons qu'il est possible de créer des estimateurs d'états robustes aux attaques ayant une connaissance parfaite du réseau électrique à attaquer, surpassant significativement les méthodes de protection présentées précédemment.

La deuxième méthode sur laquelle nous nous concentrons et qui vise à protéger les modèles basés sur l'apprentissage profond est la détection d'échantillons adverses. Inspiré par le concept des canaux de réjection, il est possible d'augmenter la fiabilité des décisions prises par les réseaux neuronaux profonds en ajoutant un détecteur au modèle initial. De multiples méthodes de détection sont disponibles aujourd'hui, mais elles reposent souvent sur un entraînement lourd et des heuristiques ad-hoc. Dans notre travail, nous utilisons des outils statistiques simples appelés les profondeurs de données (data-depth) pour construire des méthodes de détection efficaces. Les profondeurs de données nous fournissent un ordonnancement des points vis-à-vis du "centre" d'un distribution référence. En d'autre termes, calculer les profondeurs de données permet de déterminer à quel point un point est "profond" dans une distribution référence, i.e., est similaire à la référence. Nous developpons deux méthodes distinctes de détections, dans deux cas spécifiques: le cas dit "supervisé", c'est-à-dire que les attaques sont fournies pendant l'entraînement du détecteur, et le cas "non supervisées", c'est-à-dire que l'entraînement ne peut s'appuyer que sur des échantillons propres.

Dans le cas supervisé, nous présentons `HAMPER`, méthode basée sur l'utilisation de la profondeur de masse du demi-espace (Halfspace-Mass depth) et la connaissance des attaques qui seront perpétrées afin de construire un détecteur d'attaques adverses. Nous présentons dans ce travail deux scénarios distincts: le scénario connaissant parfaitement l'attaque perpétrée, et le scénario aveugle aux attaques. Dans le deux cas, notre méthode permet de mieux détecter les exemples adverses, tout en necessitant un temps et des ressources similaires aux autres attaques. De plus, dans le cas d'attaques ayant un connaissance parfaite du modèle à attaquer et de la défense choisie, les performances de notre détecteur surpassent les performances des méthodes état-de-l'art.

Dans le cas non-supervisé, nous présentons `APPROVED`, méthode basée uniquement sur l'utlisation de la profondeur intégrée pondérée par le rang (Integrated Rank-Weighted depth) appliqué à la sortie d'un classifieur profond (c'est-à-dire aux

logits). La méthode que nous proposons présente plusieurs avantages: elle est rapide, efficace, et surpasse les autres méthodes état-de-l'art en terme de performances. De plus, elle présente, comme toutes les depths, l'avantages de ne pas être dérivable, ce qui rend plus compliqué le travail des agents malveillants dont la méthode de génération d'images est basée sur l'utilisation des gradients et la totale la connaissance de la défense. Finalement, dans le cas d'attaquant ayant la connaissance parfaite de la défense, et non basée sur les gradients, notre méthode surpasse les autres méthodes état-de-l'art.

En résumé, au cours de cette thèse, nous avons développé différentes méthodes de protection d'un modèle basé sur l'apprentissage profond face à de potentiels agents malveillants, basées sur des outils provenant de la Géométrie de l'Information et des statistiques. Nous avons proposé une méthode qui améliore la robustesse d'un modèle basée sur l'utilsation de la mesure de Fisher-Rao, dans le cas de la classification d'images, et dans le cas de la régression nécessaire au développement des estimateurs d'états pour les Smart Grids. Nous avons également proposé deux méthodes de détection des attaques adverses. Bien que différentes pour chaque méthode, elles sont toutes deux basées sur les profondeurs de données (data-depths), et chacune s'applique à l'un de ces deux cas d'étude: le cas supervisé, où la défense a une connaissance partielle ou totale de l'attaque qu'il va subir, et le cas non-supervisé, où la défense n'a aucune connaissance sur l'agent malveillant.