STIET PHilosophiae Doctor

Department of Electrical, Electronics and Telecommunications Engineering and Naval Architecture, University of Genoa

Aitek S.p.A.

Institute of Electronics, Computer and Telecommunication Engineering, Italian National Research Council

Mathematical Methods for eXplainable and Reliable Machine Learning in Trustworthy Artificial Intelligence

Alberto Carlevaro

Supervisor
Mario Marchese, DITEN

Co-Supervisors
Maurizio Mongelli, CNR-IEIIT
Stefano Delucchi, Aitek S.p.A.

External Reviewers
Teodoro Rafael Alamo Cantarero, Unveristy of Seville
Bruce Nagy, NAWCWD U.S. Department of Defense

Academic Year 2022/2023

Al nuovo che verrà,
soprattutto alla mia famiglia
che con l'arrivo di Massimo Mirco
è davvero un po' più nuova.

Acknowledgements

E' arrivato il momento di partire. Questa breve frase ha sempre dato inizio alle lettere che ho scritto in questi anni come capo scout per i ragazzi che stavano per prendere la "partenza", decidendo cosa ne sarebbe stato del loro futuro. Quando scrivevo quella frase significava che era giunta l'ora di mettersi a pensare a tutte le esperienze, gli istanti, i ricordi che avevo vissuto insieme al destinatario della mia lettera. Oggi quel destinario sono proprio io e mi devo sentire pronto a riflettere su tutta la strada che mi ha portato qui a scrivere questa tesi e a spendere queste parole. Devo dire che l'analogia con la mia vita da scout si adatta molto bene al percorso di dottorato: un po' come quando parti per la route con lo zaino che ti schiaccia e il passo pesante e pensieri misti tra l'ansia di vedere alba e tramonti nuovi e il terrore che piova tutta la settimana. Piano piano poi ti rendi conto che lo zaino diventa più leggero e il passo più svelto ma non perché la strada è meno in salita ma piuttosto perchè le tue gambe hanno imparato a conoscere la montagna. Così nella ricerca, articolo dopo articolo cominci davvero a trovare la tua strada in mezzo a tutte quelle idee e inizi a dire la tua, sempre tra l'orgoglio e il dubbio di quello che hai scritto. E ovviamente il sentiero continua ad andare su e giù, tra articoli non accettati e la mail di Scholar che ti dice che finalmente la tua ricerca è stata citata. Ma come al solito i panorami che ti ricordi meglio sono quelli più inaspettati, quelli che quando ci pensi ti sembra di averli visti solo in cartolina: la California, la Spagna, la Sardegna e i tanti altri posti che ho avuto la fortuna di poter visitare e vivere grazie alla mia ricerca e soprattutto immensamente grazie a chi l'ha resa possibile. Già, perchè ho imparato che la ricerca non si fa da soli e come quando ti dividi i pali dal resto della tenda, ti rendi conto che avere qualcuno che rilegge e migliora il tuo abstract ti fa sentire più leggero. Ma affinchè tutto questo abbia senso, c'è sempre bisogno di qualcuno che ti stia a sentire quando inizi a raccontare e che ti supporti quando vedi che le cose non vanno come vorresti. E' decisamente a loro che vanno i miei ringraziamente più sentiti. Soprattutto a chi mi ha seguito nella mia avventura più grande. Non lasciandomi andare alle nostalgie, ho davvero usato queste righe per pensare a tutto quello che ho fatto in questi anni e, senza risultare banale o esagerato, dovrei dedicare un intero capitolo della mia tesi per ringraziare singolarmente tutti quelli che hanno fatto parte del mio cammino e che mi hanno permesso di crescere professionalmente e soprattutto caratterialmente. Spero davvero di avervi fatto capire che, ognuno a proprio modo, è stato importante per me.

Credo che sia arrivato davvero il momento di partire, grazie ancora per il cammino

Albi

This Ph.D. program was granted by Aitek S.p.A. and CNR-IEIIT.

Abstract

This thesis presents contributions mainly in the field of trustworthy AI, with a focus on mathematical methodologies developed to evaluate the problem of making a machine learning algorithm reliable and controllable.

The common thread in all my research has been the goal of finding a so-called *safety region* in the input space of an inference model that allows providing probabilistic guarantees on the output of the model and tools to control the prediction. The idea of safety region fits well with the task of classification in machine learning: the goal is to classify instances into well-defined and closed envelopes, respecting some probabilistic performance or guarantees. So my research started from a thorough and accurate review of the main classification algorithms in machine learning, from support vector machines to neural networks via rule-based models as well. But the best algorithm I found to achieve my purpose was Support Vector Data Description (SVDD), an established algorithm for outlier detection whose main purpose is to enclose target data within a sphere with a center and radius learned from the data distribution. The choice of such an algorithm for defining the safety region is quite trivial and supportable: SVDD allows a closed region to be defined in the input space and also provides a radius that can easily control the shape of the classification boundary to "inflate" or "deflate" it according to the performance objective. Starting from a totally data-driven definition of safety region, with only empirical (but effective) performance guarantees, I moved to a more mathematical definition, placing my idea of safety region within the framework of probabilistic scaling. This technique, in the state of the art of order statistics, provides a clear and indisputable way to obtain probabilistic guarantees on the safety region. Here, moreover, I applied the idea of safety region to a broader class of classifiers, called *scalable classifiers*, i.e., classification models that all share a scalable parameter in the classifier's predictor definition that can be appropriately adjusted to obtain the desired guarantees for the safety region and I also specialized these concepts into *exponential distributions* that allow special properties of safety regions. This allows to extend the concepts developed in Chapter 3 from SVDD to any kind of machine learning classifier. In particular, I introduced new algorithms both to control performance in classification and to obtain probabilistic guarantees of the safety region. Performance control was achieved by minimizing the misclassification error, reducing the number of false positives or false negatives or both, depending on the

application. On the other hand, probabilistic guarantee has been shown mathematically to be effective. Both concepts, however, can be applied to real-world problems to achieve safety in cyber-physical systems applications, such as vehicle platooning monitoring, DNS tunneling detection, and type-2 diabetes disease prediction, just to name a few tested applications of my methods.

However, before getting good results in my research, several ways were tried. Another line of research for defining the safety region was the use of *conformal prediction*, a new but well-established theory for evaluating conformity in machine learning algorithm performance. In this case, the idea behind conformal prediction is that it is possible to correctly calibrate an algorithm to obtain marginal probability coverage that the desired output of the model is as expected. In this field, it is necessary to define a real-valued function, called *score function*, that encodes the characteristics of the model and calibrate the algorithm to the result of evaluating that function on a calibration set. This line of research is getting good prospects and is one of the lines I will follow in my future work.

But reliability is not enough to make AI totally trustworthy. In fact, controllability is another crucial aspect to consider. From this point of view, I focused on studying and developing new techniques to control the output of a classification algorithm. This was done in the spirit of *counterfactual explanation*, a fairly new but already state-of-the-art eXplanaible AI technique. The idea of counterfactual explanations is that it is possible to minimally change the input parameters of a machine learning algorithm so as to change the prediction results. In the sense that will be explained in the chapter dedicated to counterfactual explanations (Chapter 9) will be clear that the expression " minimal change" refers to the idea of minimizing a specific cost function between the actual input and the desired one. My contribution in this topic lies in the development of a counterfactual approach based on SVDD, totally in line with the idea of safety region investigated in the first part of my research. The proposed approach was first attempted to be solved completely analytically, but then, given the complexity of the task, a numerical solution based on random sampling techniques was developed. The algorithm, again, was applied to real-world application problems, such as crowd control in subways. This topic, however, allows for more exploration, for example by merging it together with the conformal framework provided by safety regions.

Finally, all the work presented in this thesis has been surrounded by explainable AI, the field of study dedicated to making AI explainable and expressible by intelligible rules. In this regard, explainable AI can also be declined in terms of controllability and reliability, thus placing all my research totally in line with this theme.

In conclusion, my thesis covered three years of research in the field of artificial intelligence, spending most of the time evaluating the problem of how to make a good machine learning algorithm from a reliable, explainable and controllable point

of view, with the hope of having really improved the body of knowledge in such a crucial aspect of Science.

Ph.D. Contribution

The topics covered in my Ph.D. were broad: I began by studying eXplainable Artificial Intelligence (XAI) methods to improve and make understandable complex cyber-physical systems, then I oriented my research on the need to define regions in the input space of a machine learning model such that it is possible to have probabilistic guarantees on the output of that model, for then moving my research toward a more "trustworthy" spirit, but passing through deep learning models for video content analysis and physics-informed machine learning. The reason for such a diverse range of topics studied in my Ph.D. stems from the fact that research always has to be split between industry and academia. Although at first glance this may be challenging and confusing, this duality has given me the opportunity to fully understand the meaning of research: it is really the feeling of discovering and learning new things while trying to create bridges between them.

Outline

All the results obtained in my research have always been guided by the motivation to give reliability and explanation to the complexity of machine learning techniques. In particular, the context underlying all my work has been the need to give the user of such algorithms the ability to understand and control them, and not just to be subjected to their output.
With this idea in mind, the text was divided into four different parts.

Part I presents the methodologies studied and used to give the definition of a safety region. A safety region can be defined as a subset of the input parameter space in which the output is guaranteed to be as expected, or the model has a high probability of producing the desired output. The definition of safety region is a crucial aspect in engineering research. The need to identify such a safety guarantee is crucial in many applications, from biological engineering (e.g., the need to predict with high accuracy the biomarkers responsible for the onset of a specific disease) to automotive engineering (e.g., controlling the parameters that accurately predict the parameters that will not crash an autonomous car) or any other application field in which a certain degree of safety must be guaranteed. In addition, my research also addressed the problem of making the search for this region easily understandable by users, that is, making the algorithms for defining the security region interpretable

and controllable. In particular, I provided new interpretations of black-box algorithms, such as SVDD, in terms of conformal prediction and rule-based models.

Part I is divided in three chapters:

Chapter 2 is devoted to the definition of a data-driven safety region built on the basis of the SVDD algorithm. This chapter presents the methodology by which SVDD was modified to adapt it as a safety region. The intuition behind the use of SVDD as a safety region is that this algorithm is capable of developing closed and restricted sets that can be controlled by a ray. This made it possible to generate safety regions that can be modified according to the needs of the specific application, such as improving classification accuracy, minimizing the number of false positives or negatives, and so on. In particular, two algorithms are presented: `RadiusReduction` allows controlling the number of misclassified points by simply moving the radius of SVDD, `ZeroFPRSVDD` iteratively executes successive SVDDs in the same region until a threshold is reached on the misclassified points. Both algorithms have found application in real data sets, as reported in the example sections.
Chapter 3 and **Chapter 4** show the combination of the state of the art Conformal Prediction technique and the need to define a safety region. Specifically Chapter 3 focuses on the use of probabilistic scaling and Chapter 4 on the properties that exponential probability distributions can have in defining a safety region. As far as my research is concerned, the most valuable contributions were found in the definition of scalable classifiers, a special family of classifiers that share the property of having a scalable parameter in their definition (such as offset in support vector machines), the definition of probabilistic safety region that mathematically establishes the concept of safety region, and nontrivial links between probabilistic scaling theory and the aforementioned theory of conformal prediction. This has been some of the most challenging work in my research and in fact some of the most fun I have had.
Chapter 5 is dedicated to the studies I did regarding the applicability of the conformal prediction framework to rule-based models. In particular, I focused on the definition of a proper score function to be used to develop conformal predictions for state-of-the-art rule-based models such as decision tree and logic learning machine.

Starting from the concept of safety region, I moved my research into the field of counterfactual explanations, which is the main topic of **Part II**. Counterfactual explanations are a relatively new topic in the field of explainable AI, and in a nutshell, this theory addresses the problem of finding the smallest changes in the input parameters of a machine learning algorithm (either classification or regression) such that the output changes. This theory has much in common with the theory of adversarial machine learning, which in fact I covered in my research, but it is not the subject of this part of my thesis. Specifically, I focused on defining a new counterfactual framework for classification, starting with a two-class SVDD. From the binary classification problem, for which an analytic solution can be found for linear kernels, I also addressed the problem of defining multicounterfactual explanations for mul-

ticlass datasets. This was the reason that prompted me to study the generalization of SVDD to the multiclass case, which is one of my most important contributions with code available on github. **Chapter 6** and **Chapter 7** respectively describe the binary and multiclass counterfactual SVDD approach.

One of the objectives of my research was to apply the theoretical methodologies I developed to real-world scenarios. This is exactly what is reported in **Part III**. In particular, **Chapter 8** reports the studies I conducted on the usability of the concept of safety region described in Part I in the context of adversarial machine learning, where the problem was to find an attacked subset of the input features through reliable methods such as SafeSVDD. **Chapters 9** and **Chapter 10**, on the other hand, deal with the application of the counterfactual framework described in Part II of my thesis to a biomedical problem (predicting and controlling the onset of type 2 diabetes) and to improving flow management in the Genoa subway (thus, an application in the smart city context).

Finally **Part IV** with **Chapter 11** reports conclusions and future works.

A detail timeline, exposed with a Gantt schema, is reported in Figure 1.

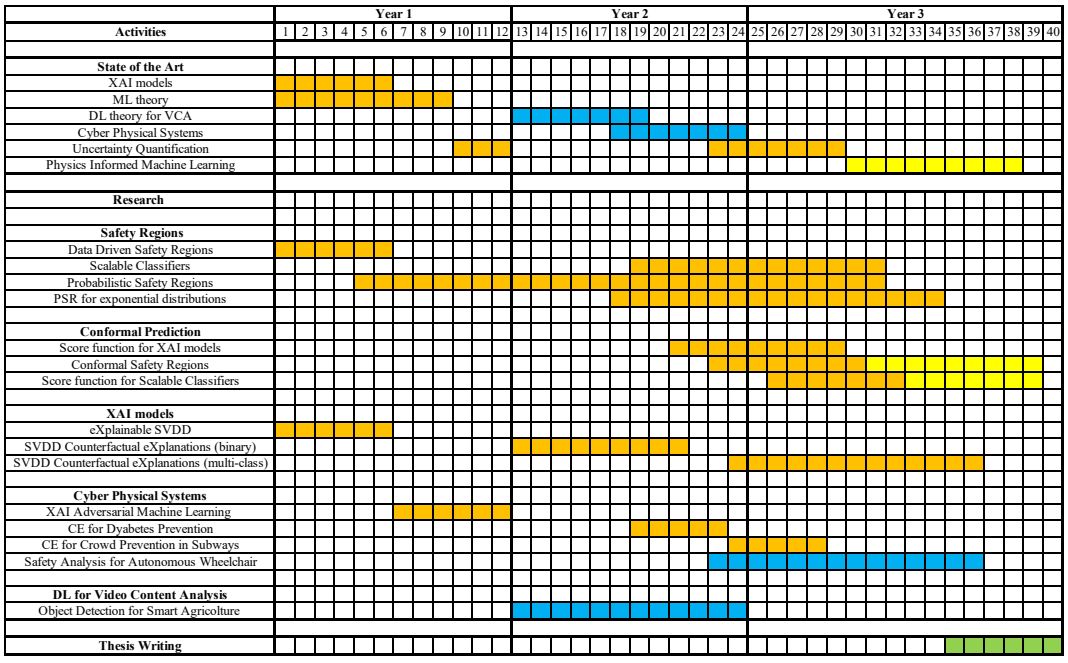


Figure 1: Gantt schema of the research activities. The activities are divide by color depending on the institution who hosted the specific research: orange CNR, blu Aitek, yellow UC Berkeley and green UNIGE.

Publications

The results shown throughout this thesis have been published in several journals and conference proceedings, although some of them are still under review.

Please check my personal page on Google Scholar.

- **Journals**

  - M. Lenatti, A. Carlevaro, A. Guergachi, K. Keshavjee, M. Mongelli, A. Paglialonga **A novel method to derive personalized minimum viable recommendations for type 2 diabetes prevention based on counterfactual explanations**, in Plos One, vol. 17, issue 11, Published: November 17, 2022, doi: 10.1371/journal.pone.0272825.
  - I. Vaccari, A. Carlevaro, S. Narteni, E. Cambiaso and M. Mongelli, **eXplainable and Reliable Against Adversarial Machine Learning in Data Analytics**, in IEEE Access, vol. 10, pp. 83949-83970, 2022, doi: 10.1109/ACCESS.2022.3197299.
  - A. Carlevaro, M. Lenatti, A. Paglialonga and M. Mongelli, **Counterfactual Building and Evaluation via eXplainable Support Vector Data Description**, in IEEE Access, vol. 10, pp. 60849-60861, 2022, doi: 10.1109/ACCESS.2022.3180026.
  - A. Carlevaro and M. Mongelli, **A New SVDD Approach to Reliable and eXplainable AI**, in IEEE Intelligent Systems, doi: 10.1109/MIS.2021.3123669.
  – Sassu, Alberto and Motta, Jacopo and Deidda, Alessandro and Ghiani, Luca and Carlevaro, Alberto and Garibotto, Giovanni and Gambella, Filippo, **Artichoke Deep Learning Detection Network for Site-Specific Agrochemicals Uas Spraying**. Available at SSRN: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4272684`.

- **Conferences**

  - Narteni, S., Carlevaro, A., Dabbene, F., Muselli, M., & Mongelli, M. (2023, August). **CONFIDERAI: CONFormal Interpretable-by-Design score function for Explainable and Reliable Artificial Intelligence**. In Conformal and Probabilistic Prediction with Applications (pp. 485-487). PMLR.
  - J. Motta, A. Sassu, A. Deidda, L. Ghiani, A. Carlevaro, F. Gambella, G. Garibotto, **A Deep Learning Artichoke Plants Identification Approach for Site-Specific UAV Spraying**, 12th International AIIA Conference: September 19-22, 2022 Palermo - Italy
  - I. Vaccari, A. Carlevaro, S. Narteni, E. Cambiaso and M. Mongelli, **On The Detection Of Adversarial Attacks Through Reliable AI**,

IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2022, pp. 1-6, doi: 10.1109/INFOCOMWKSHPS54753.2022.9797955.

- Marta Lenatti, Alberto Carlevaro, Karim Keshavjee, Aziz Guergachi, Alessia Paglialonga, Maurizio Mongelli, **Characterization of Type 2 Diabetes using Counterfactuals and Explainable AI**, 32nd Medical Informatics Europe Conference (MIE2022), May 27th - 30th 2022, Nice.
- Carlevaro A., Mongelli M. (2021) **Reliable AI Through SVDD and Rule Extraction**. In: Holzinger A., Kieseberg P., Tjoa A.M., Weippl E. (eds) Machine Learning and Knowledge Extraction. CD-MAKE 2021. Lecture Notes in Computer Science, vol 12844. Springer, Cham. `https://doi.org/10.1007/978-3-030-84060-0-10`
- Maurizio Mongelli, Alberto Carlevaro, Marta Lenatti, Martina Mammarella, Marco Muselli, Sara Narteni, Vanessa Orani, Alessia Paglialonga, Fabrizio Dabbene (2022) **eXplainable and Reliable AI Approaches to Trustworthy AI**, II Convegno Nazionale CINI sull'Intelligenza Artificiale (Ital-IA 2022).
- Mongelli M. Paglialonga A., Lenatti M., Orani V., Carlevaro A., Narteni S., Muselli M., Dabbene F., **AI & Health: Methods and Applications**, II Convegno Nazionale CINI sull'Intelligenza Artificiale (Ital-IA 2022).

- **Codes**
  - `MultiClassSVDD`, `https://github.com/AlbiCarle/MultiClass_SVDD`.
  - `CounterfactualSVDD`, `https://github.com/AlbiCarle/CounterfactualSVDD`.
  - `ZeroFPRSVDD`, `https://github.com/AlbiCarle/ZeroFPR_SVDD`.
  - `MUCH`, `https://github.com/AlbiCarle/MUCH.git`

- **Dataset**
  - `Adversarial Machine Learning Dataset`, `https://www.kaggle.com/datasets/cnrieiit/adversarial-machine-learning-dataset`.

- **Under submission or current works**
  - Carlevaro, A., Alamo, T., Dabbene, F., & Mongelli, M. (2023). **Probabilistic Safety Regions Via Finite Families of Scalable Classifiers**. arXiv preprint `https://doi.org/10.48550/arXiv.2309.04627`
  - Carlevaro, Alberto, et al. **CONFIDERAI: a novel CONFormal Interpretable-by-Design score function forExplainable and Reliable Artificial Intelligence**. arXiv preprint `arXiv:2309.01778` (2023).

  - Carlevaro, Alberto, Lenatti, Marta, Paglialonga, Alessia, Mongelli, Maurizio, 2023, **Multi-Class Counterfactual Explanations using Support Vector Data Description**, `https://www.techrxiv.org/ndownloader/files/39495529`
  - Sentient Spaces in the ECSEL FRACTAL Project: the Intelligent Totem Use Case Demonstrators. Gianluca Brilli, Alberto Carlevaro, Chiara Garibotto, Jacopo Motta, Vittoriano Muttillo, Giacomo Valente, Damiano Vallocchia, Paolo Burgio, **Sentient Spaces in the ECSEL FRACTAL Project: the Intelligent Totem Use Case Demonstrators**, submitted to Microprocessors and Microsystems.
  - Carlevaro, Alberto; Lenatti, Marta; Paglialonga, Alessia; Mongelli, Maurizio (2023). **A Counterfactual-Based Approach to Prevent Crowding in Intelligent Subway Systems**, submitted to IEEE Intelligent

Research Projects

I have actively participated in the following research projects

- **REXASI-PRO**, Horizon 2020 (October 2022 -).
  - I am currently working on WP2 - Requirements for the development of a reliable and explainable AI framework. In addition, I am working on defining the functional architecture of the project's use cases (which have as their final goal the development of a fully autonomous wheelchair) and related risk analysis.
- **More Than This**, POR Liguria 2019.
  - I worked on the extraction of data from the Genoa subway simulator (implemented by STAM srl) and implemented a classical and original machine learning algorithm for the prediction of train waiting platform crowding. In addition, I created a questionnaire on the goodness of fit of the results obtained from the model that will be used as a statistical benchmark for a future publication on AI techniques for smart cities.
- **Comp4Drones**, Horizon 2020.
  - In this project, I actively worked on both the implementation and exploitation of the results. In particular, I contributed to the definition of an object detector (a feature pyramid network, a special type of convolutional neural network) for detection of artichoke plants from a UAV spraying drone. The results obtained, in collaboration with other researchers, were published in the 12th AIIA International Conference and in the Smart Agriculture technology Journal published by Elsevier.
- **NextPerception**, Horizon 2020 (May 2020 - April 2023).

  - I worked on extracting and labeling data and writing materials for some of the project deliverables.

- **Fractal**, Horizon 2020 (September 2021 - August 2023).

  - I worked on extracting and labeling data and writing materials for some of the project deliverables.

- **CASTORE**, (2020-2021).

  - This was the first project I worked on. I followed it from the data acquisition part, working directly with the use case provider and all the other partners. I managed the data, doing statistical analysis on the relationship between the port of Genoa and the vehicular traffic in its proximity. Then I built an autoregressive model, later made explainable, based on the extracted data (both from the port and city traffic) for predicting bus inter-arrival times at bus stops of interest (30 rule based models) working in the perimeter area of the port.

Peer Review

I have been reviewing more than 50 papers for several scientific journals, such as IEEE Access, IEEE Intelligent Systems, and Springer Nature. See my profile on Web of Science `https://www.webofscience.com/wos/author/record/2472385`.

Acronyms

Here below the reader can find a list of the most used acronyms or abbreviations in my Thesis.

AI	Artificial Intelligence
CE	Counterfactual Explanation
CP	Conformal Prediction
DL	Deep Learning
DT	Decision Tree
DNS	Domain Name System
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
KKT	Karush-Kuhn-Tucker
LLM	Logic Learning Machine
LR	Logistic Regression
ML	Machine Learning
NN	Neural Network
PSR	Probabilistic Safety Region
RBF	Radial Basis Function
ROA	Region Of Attraction
SC	Scalable Classifier
SV	Support Vector
SVDD	SV Data Description
TC-SVDD	Two Class SVDD
MC-SVDD	Multi Class SVDD
SVM	Support Vector Machine
SSVM	Safe SVM
TN	True Negative
TP	True Positive
XAI	eXplainable AI

Table 1: List of Acronyms

Background Recap

This introductory discussion is devoted to illustrating the methodologies and useful tools I have used in my research. The purpose of this section is not to go into detail, but to provide the reader with the background necessary to understand the discussions that will follow.

Recap. Basic Standard ML Doctrine

Given a

- training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$,
- a model f,
- a loss function $l(y, \hat{y})$,
- an optimizer,

the goal of a supervised machine learning (ML) setup is to make an inference $\hat{y}$ on new data $\boldsymbol{x}$ as follows: $\hat{y} = f(\boldsymbol{x}; \hat{\boldsymbol{w}})$, where $\hat{\boldsymbol{w}}$ are the learned parameters.

As for notation needs, in the future we will indicate, when not misleading, the enumeration of an index with a pair of square brackets, e.g.

$$[n] \doteq i = 1, \dots, n.$$

The basic approach is to find the optimal $\boldsymbol{w}$ for our optimization problem,

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(\boldsymbol{x}_i; \boldsymbol{w})).$$

However, in real world it is infeasible to handle directly with the minimization of the loss, since the uncertainty brought by data must be taken into account. We can model the real world using a probability distribution $P(\boldsymbol{x}, y)$ underneath all the available data $\boldsymbol{x}_i,\ i \in [n]$, and aim to minimize the expectation of our loss function with respect to this probability function:

$$\mathbb{E}_{\boldsymbol{x},y}[l(y, f(\boldsymbol{x}; \boldsymbol{w}))].$$

Also in this case we can deal with other issues, for example the over-fitting of $\hat{\theta}$. To avoid this problem, it is sufficient (in ost of the cases) to add a regulizer during the training

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \underbrace{\left[\frac{1}{n} \sum_{i=1}^{n} l(y_i, f(\boldsymbol{x}_i; \boldsymbol{w})) + R(\boldsymbol{w}) \right]}_{L_{\boldsymbol{w}}},$$

where $R(\boldsymbol{w})$ is chosen accordingly with the loss function (a common set is for example $R(\boldsymbol{w}) = \lambda \left\| \boldsymbol{w} \right\|_2^2$).

To build a model, it is necessary to set up properly the hyperparameters. This is usually done splitting the training set into a *proper training set* and in a *validation* or *calibration* set.

When everything has been set up, the training is carried on solving the minimization problem through *gradient descent*, an iterative approach to optimization (with the spirit of Newtons method) that seeks the local optima taking repeated steps in the opposite direction of the gradient around the current point:

$$\hat{\boldsymbol{w}}_{t+1} = \boldsymbol{w}_t + \eta(-\nabla_{\boldsymbol{w}} L_{\boldsymbol{w}}(\boldsymbol{w}_t)).$$

The general framework for the definition of a machine learning model follows basically what exposed above, but it is only the minimal idea of the process. In the following, I briefly explain two of the techniques that I used the most during my PhD, that both collocate in the machine learning framwork.

Support Vector Data Description

Support Vector Data Description (SVDD) [10] is a good example of the general framework of a machine learning model described above. In this case the idea is to enclose data in the smallest hypersphere minimizing the variance and maximising the information of the data structure, i.e. drawing an hypersphere that contains as much points as possible minimizing the volume of the sphere. Modifications of this algorithm, that I treated along my PhD studies allow to perform classification of specific classes of target objects, i.e. it is possible to identify a region (a closed boundary) in which objects which should be rejected are not allowed. The algorithm, explained here at a glance, was first published in [10], addressing the problem of one class classification for outlier detection.

[0]Kools, J.: 6 functions for generating artificial datasets (`https://www.mathworks.com/matlabcentral/fileexchange/41459-6-functions-for-generating-artificial-datasets`), MATLAB Central File Exchange.

Let $\{\boldsymbol{x}_i\}, i \in [n]$ with $\boldsymbol{x}_i \in \mathbb{R}^d$, $d >= 1$, be a training set for which we want to obtain a description. We want to find a sphere (a hypersphere) of radius R and center $\mathbf{a}$ with minimum volume, containing all (or most of) the data objects.

$$\begin{aligned} \min_{R,\mathbf{a}} \quad & F(R,\mathbf{a}) = R^2 \\ \text{s.t.} \quad & ||\boldsymbol{x}_i - \mathbf{a}||^2 \leq R^2 \quad \forall i \in [n]. \end{aligned}$$

Introducing slack variables $\xi_i \geq 0$ to relax the problem and finding the Lagrangian under the Karush-Kuhn-Tucker conditions, a new optimization problem for the Lagrange multipliers is defined

$$\begin{aligned} \max_{\alpha_i} \quad & L = \sum_i \alpha_i(\boldsymbol{x}_i \cdot \boldsymbol{x}_i) - \sum_{i,j} \alpha_i \alpha_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j) \\ \text{s.t.} \quad & \sum_i \alpha_i = 1, \\ & 0 \leq \alpha_i \leq C, \quad \forall i \in [n], \end{aligned}$$

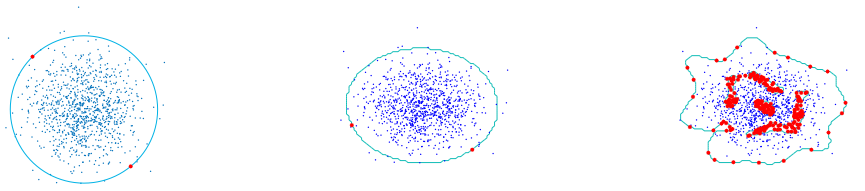
and its solution allow to retrieve the center

$$\mathbf{a} = \sum_i \alpha_i \boldsymbol{x}_i$$

and the radius, computed as the distance between the $\mathbf{a}$ and any input *support vector* $\boldsymbol{x}_s$ with corresponding Lagrange multiplier such that $0 < \alpha_s < C$

$$R^2 = ||\boldsymbol{x}_s - \mathbf{a}||^2 = (\boldsymbol{x}_s \cdot \boldsymbol{x}_s) - 2\sum_i \alpha_i (\boldsymbol{x}_s \cdot \boldsymbol{x}_i) + \sum_{i,j} \alpha_i \alpha_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j).$$

To allow a more flexible description, all the dot products $(\boldsymbol{x} \cdot \mathbf{y})$ can be substitute with suitable kernel functions $k = k(\boldsymbol{x}, \mathbf{y})$ satisfying Mercer's Theorem [139].



(a) Linear Kernel (b) Polynomial Kernel (c) Gaussian Kernel

Figure 2: SVDD with different kernels.

Generalization of the above procedure can be found in literature, specifically

- [19] provides an extension to the binary classification problem when there is a target class to be enclosed inside the sphere and a negative class to be taken outside ;

- [36] generalizes the problem at two target classes, introducing the TC-SVVD algorithm (Two Class SVDD);
- in a my recent work under review, [169], I generalized the algorithm to the multi-class case, i.e. MC-SVDD. A Matlab repository with the online code is also available at `https://github.com/AlbiCarle/MultiClass_SVDD`.

Logic Learning Machine

The Logic Learning Machine (LLM)[1] is a rule-based model

if <*premise*> **then** <*consequence*>

implemented on the basis of the Switching Neural Networks [23].

Given an input example space for (binary) classification, $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N} \in \mathcal{X} \times \mathcal{Y}$, the LLM model learns a classifier, $g : \mathcal{X} \longrightarrow \mathcal{Y}$, which can be described by a set of decision rules $\mathcal{R} = \{r_k\}_{k=1}^{N_r}$, each expressed with the classical form *If* <*premise*> *then* <*consequences*>. The <*premise*> constitutes the antecedent of the rule and is a logical conjunction ($\wedge$) of conditions c_{l_k}, with $l_k = 1_k, \ldots, N_k$, on the input features making up any sample $\boldsymbol{x}_i \in \mathcal{T}$. The <*consequence*> expresses the output class of the decision rule. Each rule $r_k \in \mathcal{R}$ can be evaluated through two useful metrics, namely covering $C(r_k)$ and error $E(r_k)$, defined as follows:

$$C(r_k) = \frac{TP(r_k)}{TP(r_k) + FN(r_k)} \quad E(r_k) = \frac{FP(r_k)}{TN(r_k) + FP(r_k)}.$$

Being $\hat{y}_i$ the class label predicted by the LLM for point $(\boldsymbol{x}_i, y_i)$, $TP(r_k)$ and $FP(r_k)$ are defined as the number of instances that correctly and wrongly satisfy rule r_k, being $\hat{y}_i = y_i$ and $\hat{y}_i \neq y_i$ respectively; conversely, $TN(r_k)$ and $FN(r_k)$ represent the number of samples $(\boldsymbol{x}_i, y_i)$ which do not meet at least one condition in rule r_k, with $\hat{y}_i \neq y_i$ and $\hat{y}_i = y_i$, respectively. By combining covering and error, the *rule relevance* $R(r_k)$ of rule r_k can be computed as:

$$R(r_k) = C(r_k) \cdot (1 - E(r_k)).$$

It is worth underlying that the LLM design process is based on an aggregate-and-separate approach [107] able to generate a set of rules that are not *disjoint*. As a result, an input sample $\boldsymbol{x}_i$ may verify multiple rules predicting the same class label and it even may cover rules predicting different outputs. When rules predicting different output labels are contemporary present in this set, class assignment is performed by computing a classification score value

$$S_{LLM}(\boldsymbol{x}, y) = \sum_{r \in \mathcal{R}_{\boldsymbol{x}}^{y}} R(r),$$

[1] `https://www.rulex.ai`

where $\mathcal{R}_{\boldsymbol{x}}^{y}$ is the subset of rules predicting y verified by the point $\boldsymbol{x}$.
A class label $\hat{y}$ is then assigned to $\boldsymbol{x}$ through the classification score, as follows:

$$\hat{y} = arg\max_{y} S_{LLM}(\boldsymbol{x}, y).$$

Contents

Chapter 1

Introduction

1.1 The need for trustworthy AI

Increasingly in recent times, the mere prediction of a machine learning algorithm is considered insufficient to gain complete control over the event being predicted. As a matter of fact, recently European Union regulated the use of artificial intelligence by the AI Act[1], the world's first complete AI law. This is a milestone for AI research, which established some pillars that can no longer be ignored. In this sense, the EU Guidelines put forward a set of seven key requirements that AI systems should meet to be considered trustworthy [154]: 1) Human agency and oversight, 2) Technical Robustness and safety, 3) Privacy and data governance, 4) Transparency, 5) Diversity, non-discrimination and fairness, 6) Societal and environmental well-being, 7) Accountability. But I would suggest that two requirements are still missing: eXplainability and Reliability. A machine learning algorithm should be considered reliable in the way it allows to extract more knowledge and information than just having a prediction at hand. In this perspective, the eXplainable Artificial Intelligence (XAI) theory plays a central role. As a matter of fact, in several contexts, qualitative information about the system is essential. Consider, for example, biological and medical problems, where the mechanisms of disease onset must be discussed with medical personnel. For this and many other reasons researchers, students and the overall scientific community in the field of artificial intelligence is making an effort in improving the body of knowledge about XAI. For a more comprehensible and safe use of learning technology.

1.2 eXplainable Artificial Intelligence

The literature in eXplainable AI (XAI) is extensive [128]. This section briefly presents a taxonomy of XAI techniques [155] in order to get the reader more inclined to fully understand the arguments presented in this thesis.

[1] `https://www.europarl.europa.eu/news/en/headlines/society/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence`.

It is first useful to begin with a general overview [32]. An efficient machine learning algorithm must achieve a good level of accuracy (i.e., it must reduce prediction errors in the system output). ML techniques can be grouped into three categories:

- Methods based on probability estimation, the most classical statistical techniques, typically based on Bayesian methods.
- Black-box methods ("black-box"), which build the model without concern for the interpretability of its inner workings, but considering only the fit of the model output to the available data.
- Intelligible methods, based on rule generation that realize an intelligible model of the system under consideration.

On the other hand, the level of interpretability of the solution provided by the method is also a crucial issue when the user requires a deep understanding of the system studied. A novel approach, called Logic Learning Machine (LLM), gives rise to models that describe an intelligible set of rules with a level of accuracy comparable to or higher than that of the best ML algorithms. LLM will be discussed in a devoted section behind.

Methods based on probability estimation: methods belonging to this class attempt to estimate the probability distribution of the examples in the training set. From this estimate, the expected risk, that is, a measure of the amount of error in the input space, is minimized. Based on the different probability function estimation assumptions, the methods in this class are divided into two groups:

- Parametric techniques, assuming a functional form for the probability distribution and finding the set of parameters that best fits the available points.
- Non-parametric techniques, estimating the probability distribution by counting patterns that are in a sufficiently small region of the input space.

Black-box methods: an alternative approach is to directly minimize the empirical cost functional, that is, the expected cost calculated on the training set examples, without estimating the probability distribution. It has been shown that this approach usually offers better performance than probability estimation methods, both in terms of accuracy and computational resources required. In this case, algorithms must retrieve a function that best describes the relationship between input and output. Depending on the class of functions in which the function is sought, different learning models can be introduced. The best known are:

- Multilayer perceptrons or neural networks (NNs): these arose to emulate the behavior of the human brain and are probably the best known and most widely used learning technique. NNs are based on a combination of elementary perceptrons, that is, devices whose output depends on a weighted sum of their inputs. The training of an NN is based on two stages:

  - recovery of optimal parameters through a gradient descent procedure.
  - Definition of the network topology (i.e., the number of perceptrons and the connections between them).

  One of the disadvantages of NNs is the need to define the network topology before training: in fact, several tests with different configurations are usually performed to find the optimal network, thus increasing the computational time.

- Support Vector Machines (SVMs): were introduced to overcome some of the disadvantages of NNs and are based on the definition of a kernel function, which is used to define a mapping of points in a larger space in which the system can be modeled by linear behavior. The optimal model is then recovered through a simple quadratic programming problem. Again, the choice of kernel function is crucial, as it can affect the quality of the solution.
  Moreover, SVDD can be considered an example of SVM since, basically, the only difference between the two methods is that SVDD performs hyperspheres rather that hyperplanes like SVM.

The main disadvantage of black-box methods is related to the inability to interpret the solution provided by the algorithm. In fact, it is usually a very complicated function of the input. Also, it is worth noting that these techniques are not suitable for dealing with categorical inputs. Although these variables can be mapped to integers, the performance of the algorithms would be greatly reduced.

Explainability: over time, researchers have sought to understand and explain the inner workings of ML models [134]. XAI approaches can help solve a number of critical problems that arise when distributing a product or making decisions based on automatic predictions, including:

- Correctness: Are we sure that all and only the variables of interest contributed to our decision? Are we sure that spurious patterns and correlations were eliminated in our outcome?

- Robustness: Are we sure that the model is not susceptible to small perturbations, but if it is, is it justified for the result? In the presence of missing or noisy data, are we sure that the model does not misbehave?

- Bias: Are we aware of any data-specific biases that unfairly penalize groups of individuals, and if so, are we able to identify and correct them?

- Improvement: How can the prediction model be concretely improved? What effect would additional training data or an improved feature space have?

- Transferability: In what concrete way can the prediction model for one application domain be applied to another application domain? What properties of the data and model should be adapted for this transferability?

- Human understandability: Are we able to explain the algorithmic mechanism of the model to an expert? Perhaps even to a layman? Is this a factor conducive to greater dissemination of the model?

- Trust: Can we trust the results proposed by the [118] model? Also, how much trust do we have in models that give wrong answers with high confidence [157]?

From the available literature [161],[128], it is possible to identify five main criteria for discriminating XAI methods, as summarized in the figure below.

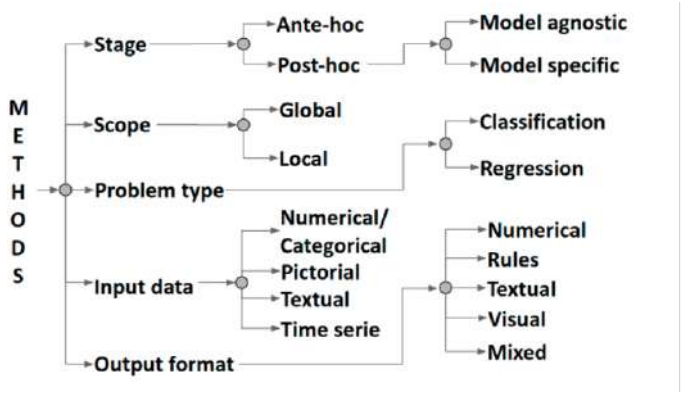


Figure 1.1: Gerarchical Classification of XAI [156].

The scope of an explanation can be global or local. In the first case, the goal is to make the entire inferential process of a model transparent and understandable. In the second case, the goal is to explain each individual inference of a model. The second dimension refers to how a method generates explanations. Ante hoc methods aim to consider the explainability of a model from the beginning and during training, to make it naturally understandable while striving to achieve good accuracy [83]. Post hoc methods keep the trained model unchanged and mimic or explain its behavior using an external explainer at the time of testing [83]. The third level of the figure refers to the type of problem. XAI methods can vary depending on the underlying problem, whether it is "classification" or "regression". Finally, the mechanisms followed by a model to classify images may be substantially different from those used to classify textual documents; therefore, the input data of a model (numerical/categorical, pictorial, textual, or time series) may play an important role in the construction of a method for explainability. An additional criterion must also be taken into account, namely the format of the output [155].
Similar to input data, different circumstances may require different output formats of explanations to be considered in a method for explainability: numerical, rule, textual, visual, or ẗmysticalẗ [128].

Type of explanations: as introduced in [114], the following types of post-hoc explanations can be considered:

- Textual explanations generate symbol-based explanations, such as natural language texts or propositional symbols that explain the intrinsic logic of the model by means of abstract concepts encapsulating high-level processes.

- Visual explanations aim to visualize the intrinsic logic of the model to facilitate understanding. These techniques are useful for obtaining information about the model's decision boundary or interactions among input features. For this reason, visual explanations are often used when targeting an audience with limited knowledge in the field of artificial intelligence.

- Local explanations focus on explaining model behavior in a specific area of interest. This means that the resulting explanations approximate the model's logic around the observation that the user wants to explain.

- Example explanations select representative observations from the training set to demonstrate how the model works. This is somewhat similar to the way humans approach explanations, namely by providing specific examples to describe a general process. Clearly, an example only makes sense if the training data must be in a form humans understand, such as pictures, whereas arbitrary numerical vectors may contain information that is difficult to retrieve.

- Explanations by simplification approximate an opaque model with a simpler one that is easier to interpret. This simple model must be flexible enough to accurately approximate the opaque model. In classification problems, this property is usually measured by comparing the accuracy of these two models.

- Feature relevance explanations aim to quantify the influence of each input variable in producing the model result. The result is a ranking of relevance scores, with higher scores associated with the most important input variable for the model. These scores provide some indication of the internal logic of the model, although they cannot provide a complete explanation.

Part I

Probabilistic Safety Regions and Conformal Prediction

Chapter 2

Data Driven Safety Regions

A ML algorithm is considered reliable if it can afford to guarantee the desired output with a low risk of error. One of the most useful but challenging techniques is to define regions in the input parameter space that guarantee such reliability. In particular, the goal is to find constraints in the input parameters such that the error in prediction is minimized. In this chapter I report the techniques I have developed on this topic during my research, which has focused primarily on classification. At a glance, given a classifier, I was interested in looking for a region in the input space that would minimize uncertainty (i.e., low levels of false positives or negatives) while keeping the "size" (i.e., the amount of information provided by the data) as large as possible.

2.1 Introduction

Improving reliability of prediction confidence remains a significant challenge in ML, as learning algorithms proliferate into difficult real-world pattern recognition applications. The intrinsic statistical error introduced by any ML algorithm may lead to criticism by safety engineers. The topic has received a great interest from industry[1], in particular in the automotive[2] and avionics [180] sectors. In this perspective, the conformal predictions framework [46] studies methodologies to associate reliable measures of confidence with pattern recognition settings including classification, regression, and clustering. The proposed approach follows this direction, by identifying methods to circumvent data-driven safety envelopes with statistical zero errors. We show how this assurance may limit considerably the size of the safety envelope (e.g., providing collision avoidance by drastically reducing speed of vehicles) and focus on how to find a good balance between the assurance and the safety space.

My work focused on a specific machine learning methods, the Support Vector Data Description, which by (its) definition is particularly suitable to define safety en-

[1] `https://www.iso.org/committee/6794475.html`
[2] `https://www.iso.org/standard/70939.html`

velops. Moreover, I added intelligible models for knowledge extraction with rules: intelligibility means that the model is easily understandable, e.g. when it is expressed by Boolean rules. Decision trees (DTs) are typically used towards this aim. The comprehension of neural network models (and of the largest part of the other ML techniques) reveals to be a hard task. Together with DT, I used LLM, which may show more versatility in rule generation and classification precision.

My work takes a step forward in these areas due to

- safety regions are tuned on the basis of the radius of the SVDD hypersphere
- simple rule extraction method from SVDD compared with LLM and DT

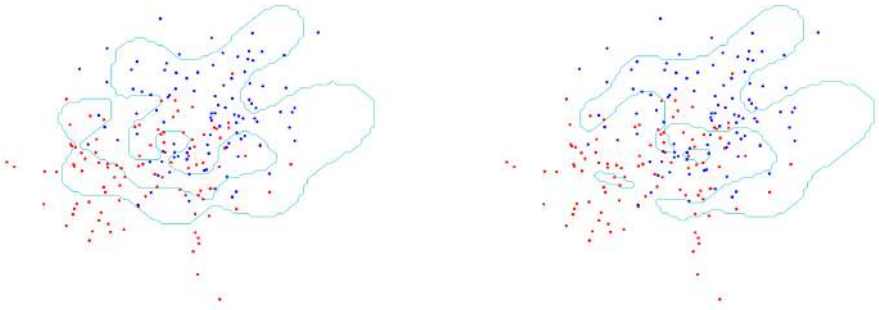
This Chapter shows the main results in this topic published in my first paper [171], specifically how to construct safety envelops from SVDD, how to make them intelligible through rules and the evaluation of the methodology in real world application datasets.

2.2 Safe SVDD

Safety regions research is a well-known task for machine learning [86, 105] and the main focus is to avoid false positives, i.e., including in the safe region unsafe points. In this section, two methods for the research of zero FPR regions are proposed: the first one is based simply on the reduction of the SVDD radius until only safe points are enclosed in the SVDD shape, the second one instead performs successive iterations of the SVDD on the safe region until there are no more negative points.

2.2.1 Radius Reduction

Since also in the transformed space via feature mapping the shape of SVDD is a sphere, it is reasonable to think that reducing the volume of the sphere the number of negative points misclassifed should reduce (see Figure 2.1).
The algorithm I implemented is based on a very simple consideration: the radius of the SVDD is reduced until a suitable predefined threshold is reached (minimize FPR or FNR, but also maximize accuracy, F1 score, etc.). The convergence of the algorithm is guaranteed, but this procedure can lead to a very small safety region. When the data set is very complex and most of the features overlap, simple radius reduction provides only safety guarantees without taking into account the volume (or area) of the region. Strictly speaking, one could obtain regions with only one or two points, satisfying the classification error threshold, but making a very bad classification. One of the main problems is that it is not possible to control the geometry of the points in the kernel space, and thus define specific criteria to stop the algorithm in advance to achieve the best trade-off between safety and usability. This is one of the reasons why I implemented dynamic error control committed by SVDD as shown in the next section.



(a) FPR=0.517 (b) FPR=0.095

Figure 2.1: Application of **Algorithm 1** on a data set of 400 points sampled from a Gaussian with mean $[1, 1]$ and variance 1, 200 target objects and 200 negative examples. The algorithm converged in 12 iterations.

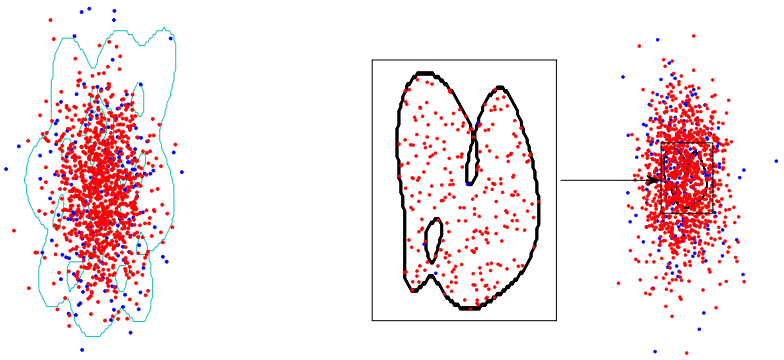
Algorithm 1 `RadiusReduction`
Dataset $\mathcal{X} \times \mathcal{Y}$ is divided in training set $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$ and test set $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$. A threshold ε is set.

---

1. SVDD-**cross-validation** on $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$
2. [**a**,R^2]=SVDD($\mathcal{X}_{tr}, \mathcal{Y}_{tr}, C_1, C_2$, param)
3. maxiter=1000;
4. i=1;
5. **while**(i<maxiter)
5.1. $R^2 = R^2$−10e-5*R^2;
5.2. **Test** SVDD on $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$
5.3. **if**(FPR< ε)
5.3.1. **return** [**a**, R^2];
5.4 **end**
6. $i = i + 1$;
7. **end**

2.2.2 Iterative SVDD for Zero Statistical Error

Here there is another algorithm to find zero FPR (resp. FPR or maximize relevance metrics) regions with SVDD. The idea is simply to perform successive SVDDs on the safe regions found with a preliminary SVDD to avoid the presence of unsafe points. Again, the convergence is achieved when a fixed number of iterations is reached or when the condition on FPR (resp. FPR or maximize relevance metrics) is satisfied.



(a) FPR=0.925 (b) FPR=0.079

Figure 2.2: Application of **Algorithm 2** on a data set of 2000 target objects sampled from a Gaussian with mean $[1, 1]$ and variance 4 and 100 negative examples sampled from a gaussian with mean $[1, 1]$ and variance 5. (a) is the first iteration of the algorithm and (b) is the convergence at the 97th iteration.

Algorithm 2 `ZeroFPRSVDD`
Data set $\mathcal{X} \times \mathcal{Y}$ is divided in training set $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$ and test set $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$. A threshold ε is set.

---

1. SVDD-**cross-validation** on $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$
2. $[\mathbf{a}, R^2]$=SVDD$(\mathcal{X}_{tr}, \mathcal{Y}_{tr}, C_{-1}, C_{+1}$, param)
3. **Test** SVDD on $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$
4. maxiter=1000;
5. i=1;
6. **while**(i<maxiter)
6.1. $\mathcal{X}_{tr_i} = \Xi(\mathcal{X}_{ts})$;
6.2. SVDD-**cross-validation** on $\mathcal{X}_{tr_i} \times \mathcal{Y}_{tr_i}$
6.3. $[\mathbf{a}_i, R_i^2]$=SVDD$(\mathcal{X}_{tr_i}, \mathcal{Y}_{tr_i}, C_{-1}, C_{+1}$, param)
6.4. **Test** SVDD on $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$
6.5. **if**(FPR< ε)
6.5.1. **return** $[\mathbf{a}^*, R^{*2}] = [\mathbf{a}_i, R_i^2]$;
6.6. **end**
7. $i = i + 1$;
end

In this case, the algorithm is more fitting: by running successive iterations of the

SVDD in the same region, the model can clearly understand the safe points from the unsafe ones. We can say, in other words, that the algorithm is cleaning the region of bad points. Unlike the previous approach, the procedure may not converge, or converge very slowly: this is mainly due to the fact that the shape of the SVDD changes with each iteration, and when the threshold is set too low the SVDD cannot work properly. This problem can be partially avoided by changing the hyperparameters of the model so that the shape fits the data better, but, reasonably, the computational time required increases dramatically.

As an example, in Figure 2.2 it is reported an example with a 2 dimensional Gaussian data set. It seems clear that the "zeroFPR" algorithm performs better safety regions than "RadiusReduction" since a new SVDD is computed at each iteration and its shape fits the data better.

2.3 eXplainable SVDD

Then I considered how to make the SVDD explainable in order to explicit the inherent logic and use the extracted rules for further safety envelope tuning as in [86]. In this part I will widely speak about Logic Learning Machine: I invite the reader to delve into details in the Background Recap preface.

Let us suppose to have an information vector $\mathbf{I}$ and to have to solve a classification problem depending on two classes $\omega = 0$ or 1. Let $\aleph = \{(\mathbf{I}^k, \omega^k), k = 1, \ldots, \beth\}$ be a data set corresponding to the collection of events representing a dynamical system evolution (ω) under different system settings ($\mathbf{I}(\cdot)$).
The classification problem consists of finding the best boundary function $f(\mathbf{I}(\cdot), \cdot)$ separating the $\mathbf{I}^k$ points in $\aleph$ according to the two classes $\omega = 0$ or $\omega = 1$. For the case of SVDD the best boundary f is simply the shape of the hypersphere. Although the shape of the hypersphere is well defined (it is enough to have a center and a radius to describe it), it is still interesting to have a rule-based shape to describe it.

2.3.1 Rules extraction from SVDD

What I did was to combine SVDD and XAI to obtain intelligible rules from the black box structure of SVDD. The derivation of intelligible rules is made as follows. After that a SVDD has been optimized, a new dataset of observations *sampled around the edge of the SVDD* is provided and the classification via SVDD is performed. The new dataset is then elaborated via a XAI algorithm; here, via the LLM, but other rule-based algorithms can be used, e.g. DT. The sampling is performed by setting a threshold $\bar{\varepsilon}$, such that the extracted observations are sufficiently close to the boundary of the trained and tested SVDD. The threshold is set a priori and depends on the dataset: given a set $\mathcal{X} = \{x_i\}_i$ of synthetic data sampled uniformly from the test set, to extract points close to the radius the quantity $t := |\ ||x_i - \mathbf{a}||^2 - R^2||\ |$ is evaluated and therefore $\bar{\varepsilon} \in (\min(t), \max(t))$. Values too close to $\min(t)$ do not allow

enough samples to be extracted while on the other hand values too close to $\max(t)$ extract too many points away from the edge of the SVDD. A good balance for the chose of ε can then be the average $(\min(t) + \max(t))/2$ or values in a neighborhood of it.

Algorithm 3 `eXplainableSVDD`
Get $\mathbf{a}^*, R^*$ from `ZeroFPRSVDD` algorithms.
Fix $\varepsilon > 0$.

---

1. **Sample** uniformly a new dataset $\mathcal{X}_{new}$ s.t.
   $x_i \in \mathcal{X}_{new} \iff \mid ||x_i - \mathbf{a}||^2 - R^2 \mid < \varepsilon$
2. **Classify** $\mathcal{X}_{new}$ in $\mathcal{Y}_{new}$ through optimal `ZeroFPRSVDD` (w.r.t. $[\mathbf{a}^*, R^{*2}]$)
3. Solve a classification problem via **LLM** w.r.t. $[\mathcal{X}_{new}, \mathcal{Y}_{new}]$
4. The LLM rules defines an explained `ZeroFPRSVDD` region $\mathcal{R}$
5. **return** $\mathcal{R}$

As in [86] I applied these rules with the goal of maximizing the number of safe points (that is the number of points in the target class) while keeping FPR (or FNR) at zero. This is possible by performing rule tuning as in [105] but SVDD allows for much more flexibility.

2.4 Examples

This section is devoted to understand how Safe SVDD works in real classification problems. First I focused on a simple example concerning the stability certification of dynamical systems through ROA [142], where I wanted to focus on the performance of rule extraction, and then I moved on a much more complex and safety relevant automotive example of cyber-physical system: the DNS tunneling detection.

2.4.1 ROA inference

The concept of *Region of Attraction* (ROA) is fundamental in the stability analysis of dynamical systems [168] and it is topical when safety of cyber physical system should be preserved with zero (probabilistic) error [105, 86].
ROA is typically derived through the level sets of Lyapunov functions but in this case I wanted to estimate ROA through SVDD: I defined the target class as the set of stable points and the negative class as the unstable ones.
Let us consider the Van der Pol oscillator in reverse time:

$$\begin{cases} \dot{x}_1 = -x_2 \\ \dot{x}_2 = x_1 + (x_1^2 - 1)x_2 \end{cases} \tag{2.1}$$

the stability region is depicted in blue in Figure (2.3). The system has one equilibrium point at the origin and an unstable limit cycle on the border of the true ROA.

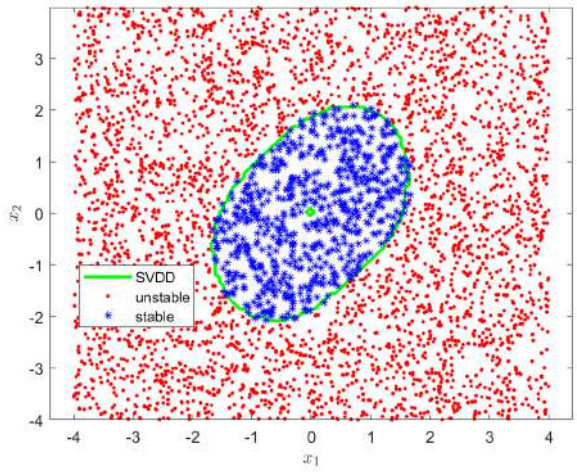


Figure 2.3: ROA of the Van der Pol oscillator. In green the SVDD shape obtained through fast-SVDD.

The simulation of the dynamical system is developed in C[3] and the dataset is composed by 300000 points (x_1, x_2) with the relative labels (+1 stable, −1 unstable). I implemented the SVDD and tested it over this dataset: the obtained results (in term of zero FNR) are good without using either Algorithm 1 or Algorithm 2 due to the good separation between the two classes. In Figure (2.3) it is shown the SVDD shape (in yellow), and the performance indices are:

$$ACC = 0.9854 \quad FPR = 0 \quad FNR = 0.0542 \tag{2.2}$$

where $ACC = \frac{TP+TN}{TP+TN+FP+FN}$ is the accuracy of the model, $FPR = \frac{FP}{FP+TN}$ is the False Positive Rate and $FNR = \frac{FN}{FP+TN}$ is the False Negative Rate.
Then a set of intelligible rules is extracted as described in Section 2.3.1 (LLM and DT) and they are tested on several extraction of different size datasets (see Figure 2.4), which are all copies of a same dataset, with the aim to profile the largest region in term of "safe points", that is the precision on the target class $\frac{TP}{TP+FP}$.
Here below, as example, the first three rules with the highest covering[4], extracted from the model through LLM:

[3] `https://github.com/mopamopa/Liapunov-Logic-Learning-Machine`
[4] The covering of a rule is the percentage of points for which that rule is true.

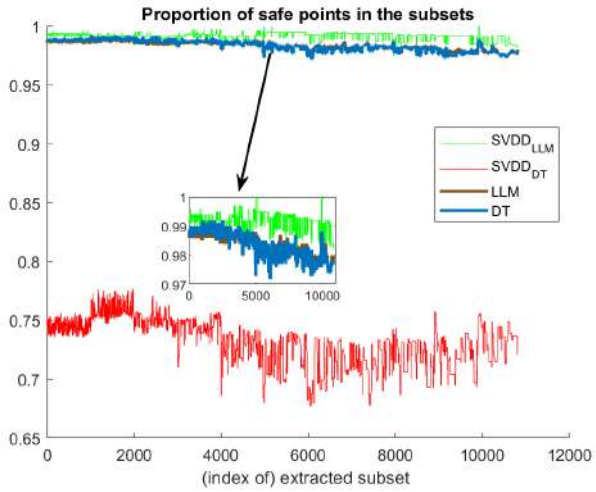


Figure 2.4: Comparison of the percentage of safe points with LLM/DT before and after SVDD, VdP example.

$$\textbf{if } (-1.6 < x_1 \leq 1.2) \wedge (-1.8 < x_2 \leq 1.8) \textbf{ then safe}$$
$$\textbf{if } x_1 \leq -1.6 \textbf{ then unsafe}$$
$$\textbf{if } (-1.6 < x_1 \leq 1.7) \wedge (x_2 \leq -1.8) \textbf{ then unsafe}$$

I made 10^3 successive extractions from the dataset (with different sizes, from 8% up to 50% of the total points): for each of them the FPR is almost zero and the precision on the target class is high, i.e. there is a good percentage of safe points. We can see that the performance of the rules extracted with DT after applying SVDD is quite inferior to the others. This is due to the fact that DT generates fewer rules than LLM and the constraint imposed by the shape of SVDD does not allow to generate rules with high coverage (i.e., small rectangles).

2.4.2 DNS tunneling

This dataset deals with covert channel detection in cybersecurity [52]; more specifically, the aim is detecting the presence of Domain Name Server (DNS) intruders by an aggregation-based monitoring that avoids packet inspection, in the presence of silent intruders and quick statistical fingerprints generation. By modulating the quantity of anomalous packets in the server, we would be able to modulate the difficulty of the inherent supervised learning solution via canonical classification schemes (Bayes decision theory, neural networks). However, our goal is to make a good classification even in the cases where the anomalous packets are very much mixed with the legitimate ones, determining the need for more precise and flexible classification methods such as SVDD.

Table 2.1: Algorithm statistics for the DNS dataset.

	FPR	% safe	# iter	# time (s)	R^2	#SV
Alg 1	0.0108	80.18	7	65.19	0.7985	61
Alg 2	0.0079	84.71	4	52.13	0.6958	31

Let q and a be the packet sizes of a query and the corresponding answer, respectively (what answer is related to a specific query can be understood from the packet identifier) and δ the time-interval intercurring between them.
The information vector of the input is composed of the statistics (mean, variance, skewness and kurtosis) of q, a and δ for a total number of 12 input features:

$$\mathbf{I} = [m_a, m_q, m_\delta, \sigma_a^2, \sigma_q^2, \sigma_\delta^2, s_a, s_q, s_\delta, k_a, k_q, k_\delta].$$

The corresponding vectors are: $\mathbf{m}$, $\boldsymbol{\sigma}$, $\mathbf{s}$, $\mathbf{k}$. High-order statistics give a quantitative indication of the asymmetry (skewness) and heaviness of tails (kurtosis) of a probability distribution, they help improve detection inference.
The training and test sets are built as follows. Let $\{(\mathbf{x}_k, \omega_k), k = 1, \ldots, \aleph\}$ be the training set ($\aleph$ is the training set size), where $\mathbf{x}_k$ is a realization of a vector containing a subset of the features $\mathbf{m}$, $\boldsymbol{\sigma}$, $\mathbf{s}$, $\mathbf{k}$ and ω_k belongs to $\{0, 1\}$ (the two classes); if the information contained in $\mathbf{x}_k$ corresponds to a DNS data exchange with tunneling: $\omega_k = 1$, $\omega_k = 0$, otherwise. An unsupervised algorithm is then used to induce the presence of a tunnel inside the data exchange characterizing a features vector. The ω label is used only as performance evaluation (test set) and it is not exploited during training.
The classification of the dataset was done through the SVDD algorithms (`RadiusReduction` and `zeroFPRSVDD`) and the results were compared with the Decision Tree algorithm and the Logic Learning Machine algorithm, as in the previous section dedicated to the ROA application. As before, our goal is to determine the largest region of parameters with no false positive (i.e. prediction of tunneling, but not tunneling in reality). To do this, we applied the two algorithms proposed in Section 2.2 to the 5000 size sample above (3000 for training and 2000 for test) using $C_1 = 1/\nu_1 N_1$, where $N_1 = \#\{\omega_k = +1\}$ and $\nu_1 = 0.01$ (i.e. we allow the acceptance of up to 1% of negative objects in the target class), $C_2 = 1/\nu_2 N_2$ where $N_2 = \#\{\omega_k = -1\}$ and $\nu_2 = 0.05$ (i.e. we allow up to 5% negative objects to be included in the classifier shape) and RBF kernel with σ determined with cross-validation. The results are shown in Table 2.1, where FPR is the usual False Positive Rate, %safe is the percentage of safe points (computed as the precision on the positive class $\frac{TP}{TP+FP}$), #iter the number of algorithm iterations, #time (s) the time in second for the convergence, R^2 the squared hyperspheres radius, #SV the number of determined support vectors.
We can observe that the `zeroFPRSVDD` in this case works well than `RadiusReduction`, achieving almost zero FPR with an acceptable large safety region.

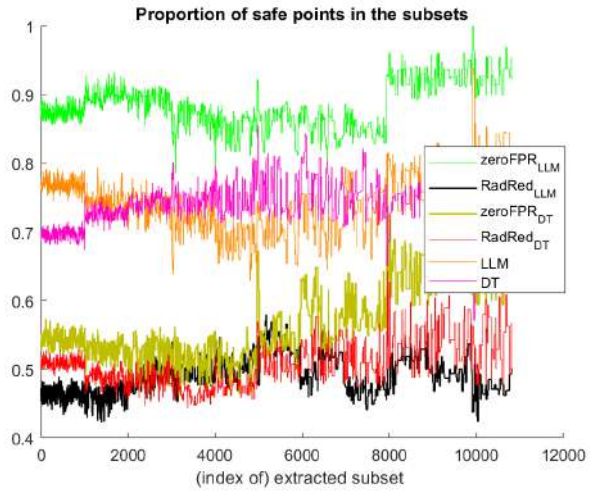


Figure 2.5: Comparison of the percentage of safe points with LLM/DT before and after SVDD, DNS-tunnelling example.

Then I tested the performances of the algorithms in different extractions of 10^3 subsets with different sizes from 8% to 50% of the total points available for test; 11×10^3 trials in total. I compared them with LLM and DT as in [86] (see Figure 2.5) and so a rules extraction has been requested. As an example, here are the first three rules for covering extracted with DT:

$$\textbf{if } m_q \leq -0.5 \textbf{ then tunnelling}$$
$$\textbf{if } -0.5 < m_q \leq 1.5 \wedge \sigma_q^2 \leq 0.4 \textbf{ then tunneling}$$
$$\textbf{if } m_q > 1.5 \wedge \sigma_q^2 \leq 0.4 \textbf{ then no tunneling}$$

Native LLM and DT are tuned according to [86, Section 4.4]. The procedure has three basic steps: (1) manually inspection of the most relevant regions for safety. (2) LLM/DT is trained with zero error when developing the rules. (3) Progressively extraction of unsafe points from the original data set until only safe points are obtained. The *native* adjective here means that the algorithms are applied directly, without SVDD interrogation. Due to its intrinsic restriction in modelling data through hyper-rectangles, see, e.g., [100], native XAI may not follow the potential tricky non-linearity that can be chased by SVDD. The analyses show that the LLM rules extracted from the SVDD model perform better classification than the other methodologies: up to 95% safe points with near-zero FPR versus only 85% for the classical LLM. The other algorithms perform sufficiently well, more than 50% of the points safe with near-zero FPR, but, as could be assumed, `zeroFPRSVDD` achieves a better safe region than `RadiusReduction`: this is probably due to the fact that `zeroFPRSVDD` fits the shape of the points better since the algorithm computes a new region at each iteration (see Fig. 2.6) while `RadiusReduction` just rigidly reduces the volume of the SVDD hypersphere until there are no more unsafe points.

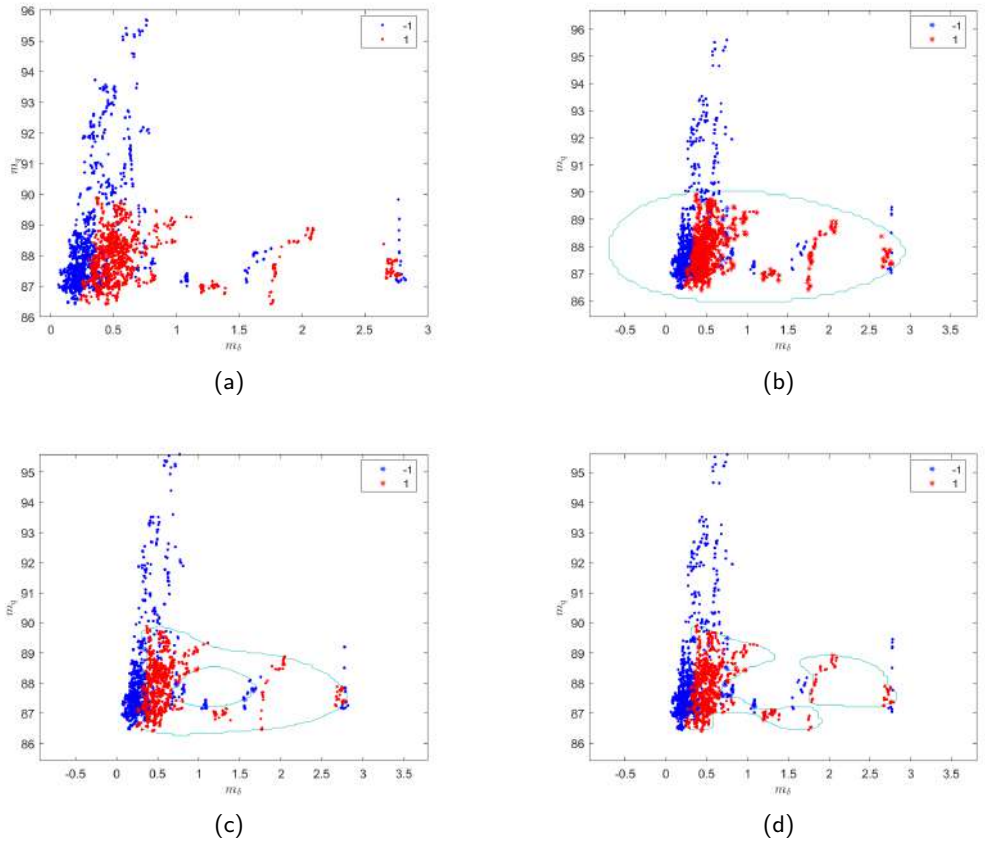


Figure 2.6: 2D graph of the evolution of the "safety region" (the red points are the tunneling ones) with `zeroFPRSVDD`: for this example I used m_δ (average interarrival time between query and answer packet over 1000 sample) and m_q (average size of query packet) as input features of the DNS tunneling dataset. The starred points are the SVs of the description, coloured referring their specific label.

Finally, I report in the following the plot (Fig.2.7) concerning the comparison between rule extraction methods with and without the sampling of the points around the edge of the SVDD region (the old algorithm is the one of [136]). It is clear that the accuracy of the classification has been improved with the new version of the `ExplainableSVDD` algorithm, thus confirming the observations reported so far.

2.5 Remarks

2.5.1 Zero statistical error

Zero statistical error refers to the discovery of the envelope, in the feature space, characterizing the presence of the points of interest of a single class only. We may refer to zero false negative (FN) when the envelope is a safety envelope as we think

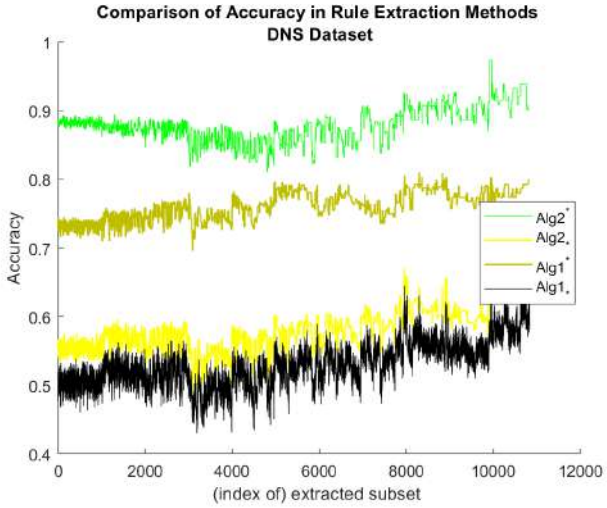


Figure 2.7: Accuracy classification of different extractions of 10^3 subsets of the DNS tunneling dataset. In the legend, the asterisked algorithms at the top (*) refer to those reported in this paper, with the rule extraction near the SVDD edge, while those asterisked at the bottom ($_*$) refer to a previous implementation of the algorithm [136]. It is clear that the accuracy of the classification is definitely improved by the new approach.

to it as the conditions for safety (e.g., no collision in a smart mobility scenario [145]); in that case, the term 'positive' means the point is outside of the safety envelope and some risk or danger may be associated to it (a collision). On the other hand, we may refer to zero false positive (FP), when we want to discover the envelope, in the feature space, in which the risk conditions are certain, namely, all the points of the envelope are anomalous or dangerous; this may be typically associated to the discovery of cyberattacks. For the sake of simplicity I have followed the zero FPR notation in both algorithm design and performance evaluation.
The term 'statistical' is associated to the fact that the metric is still based on measurements performed on the data available; it is not certain as in the formal logic perspective, which is, in turn, a way to certify safety. The two worlds (machine learning and formal logic), however, may be put in contact; recent studies are dedicated to the formal verification of neural networks [5] and the safety envelope, with zero statistical error property, may be the driver for further formal logic validation [106].

2.5.2 Data at production stage

Results shown in the figures correspond to a validation set, different from the training and test sets used in the cross validation of the algorithms. Such a validation set

[5] https://www.easa.europa.eu/sites/default/files/dfu/EASA-DDLNConcepts-of-Design-Assurance-for-Neural-Networks-CoDANN.pdf

would correspond to the production set (i.e., once the machine learning model is deployed at run time on the "production line", without further re-training), under the assumption that the (unknown) probability distribution generating the data is the same at training and production stages.
The hypothesis may be reasonable or not, depending on the specific application scenario.
In the presented ROA case, the dynamical system is fixed, not affected by noise and no differences are to be considered between training and production stages. Either any variation in the dynamic equations or any environmental noise may be considered during the training phase.
In the DNS case, raw data (from which feature samples are built) derive from the monitoring of a DNS server over a week period, in which traffic variations do not imply significant variations of the machine learning models (training and test are divided in the proportion of 50%) [53].

Chapter 3

Probabilistic Safety Regions

This part of my work is devoted to give a mathematical foundation of the concept of safety region introduced in Chapter 2. The development of this theory has been a hard task, made of trials and errors, which eventually led to the writing of a theorem proving the probabilistic guarantees of safety regions. It is worth noting that I started my Ph.D. with an approximate definition of a safety region, constructed as a simple variation of a specific algorithm, and in the end I developed a general, mathematically founded and clearly modeled definition of a probabilistic safety region. This work has definitely been the common thread throughout my Ph.D. course, which will surely lead to further research in the future.

3.1 Introduction

The theory presented in this section has two main contributions: the definition of scalable classifiers and the concept of probabilistic safety region. Both introduce a rather new level of knowledge to the field of machine learning and, more specifically, to the field of statistical learning. Scalable classifiers, as will be emphasized below, define a new class of classifiers that share the property of having a scalable parameter that can be adjusted to control the classification boundary. The new flexibility provided to the model allows it to provide probabilistic guarantees as in the framework of order statistics, leading to a natural definition of a safety region, which, due to its conformal property, can be called probabilistic.

3.1.1 Notation and order statistics concepts

Given an integer n, $[n]$ denotes the integers from 1 to n. Given $x \in \mathbb{R}$, $\lfloor x \rfloor$ denotes the greatest integer no larger than x and $\lceil x \rceil$ the smallest integer no smaller than x. The set of non-negative reals is denoted $\mathbb{R}_+$. Given integers k, n, and parameter $\varepsilon \in (0, 1)$, the Binomial cumulative distribution function is denoted as

$$\mathbf{B}(k; n, \varepsilon) \doteq \sum_{i=0}^{k} \binom{n}{i} \varepsilon^i (1-\varepsilon)^{n-i}.$$

$\Pr\{\cdot\}$ denotes the probability operator.

The following definition is borrowed from the field of order statistics [75, 166].

Definition (Generalized Max)**.** Given a collection of n scalars $\Gamma = \{\gamma_i\}_{i=1}^n \in \mathbb{R}^n$, and an integer $r \in [n]$, we denote by

$$\max^{(r)}(\Gamma)$$

the r-smallest value of Γ, so that there are no more than $r-1$ elements of Γ strictly larger than $\max^{(r)}(\Gamma)$.

To construct $\max^{(r)}(\Gamma)$ it is sufficient to order the elements of Γ as $\{\gamma_{(1)}, \gamma_{(2)}, \ldots, \gamma_{(n)}\}$ so that

$$\gamma_{(1)} \geq \gamma_{(2)} \geq \cdots \geq \gamma_{(n)}.$$

Then, we let $\max^{(r)}(\Gamma) \doteq \gamma_{(r)}$.
The following result, see Property 3 in [75], states how to obtain a probabilistic upper bound of a random scalar variable by means of the notion of generalized max. This result has been used in the context of uncertainty quantification [166] and chance-constrained optimization [97, 165].

Property 3.1. Given $\varepsilon \in (0,1)$, $\delta \in (0,1)$ and $r \geq 1$, let $n \geq r$ be such that

$$\mathbf{B}(r-1; n, \varepsilon) \leq \delta. \tag{3.1}$$

Suppose that $\gamma \in \mathbb{R}$ is a random scalar variable with probability distribution $\mathcal{W}$. Draw n i.i.d. samples $\{\gamma_i\}_{i=1}^n$ from distribution $\mathcal{W}$. Then, with a probability no smaller than $1-\delta$,

$$\Pr_{\mathcal{W}}\left\{\gamma > \max^{(r)}(\{\gamma_i\}_{i=1}^n)\right\} \leq \varepsilon.$$

Moreover, the following corollary provides a way to relate the choice of r and the number of samples n by introducing the value $\beta\varepsilon n$.

Corollary 3.2. *Let $r = \lceil \beta\varepsilon n \rceil$, where $\beta \in (0,1)$, and define*

$$\varkappa \doteq \left(\frac{\sqrt{\beta} + \sqrt{2-\beta}}{\sqrt{2}(1-\beta)}\right)^2.$$

Then, inequality (3.1) *is satisfied for*

$$n \geq \frac{\varkappa}{\varepsilon} \ln \frac{1}{\delta}. \tag{3.2}$$

Specifically, the choice of $\beta = 0.5$ leads to $r = \left\lceil \frac{\varepsilon n}{2} \right\rceil$ and $n \geq \frac{7.47}{\varepsilon} \ln \frac{1}{\delta}$.

3.2 Scalable Classifiers and Probabilistic Safety Regions

Cyber-physical systems may pose the end user to situations in which it is necessary to distinguish between what is "safe" (S) and what i "unsafe" (U). And these events can affect the system with different grade of probability. It is then clear than an appropriate control of such a situations is considered to be necessary to have a full trustworthy ML model. In this spirit, that is a generalization of the concepts exposed in the previous Chapter, my research focused on defining a *safety region*, i.e. a region $\mathcal{S}$ of the feature space $\mathcal{X}$ for which we have a guarantee that the probability of unsafe is not larger than a given *risk* level $\varepsilon \in (0, 1)$.
More formally, we consider a probabilistic framework, and assume that the observations come from a fixed probability distribution. Then, for a given *risk* level $\varepsilon \in (0, 1)$, we are interested in constructing a *Probabilistic Safety Region* (PSR), denoted by $\mathcal{S}_\varepsilon$, satisfying

$$\Pr\Big\{\boldsymbol{x} \hat{=} \mathrm{U} \text{ and } \boldsymbol{x} \in \mathcal{S}_\varepsilon\Big\} \leq \varepsilon, \tag{3.3}$$

that is, a PSR region $\mathcal{S}_\varepsilon \subset \mathcal{X}$ represents a set such that the probability of observing the event $\boldsymbol{x} \hat{=} \mathrm{U}$ conditioned to the event $\boldsymbol{x} \in \mathcal{S}_\varepsilon$ is lower or equal than ε.
To relate this theory with classical ML framework, a special (but rather general) class of classifiers is introduced. These *scalable classifiers* (SCs) are classifiers whose formulation can be made to explicitly depend on a *scaling parameter* $\rho \in \mathbb{R}$. The parameter ρ allows to dynamically adjust the boundary of the classification: a change in ρ causes a changing in the classifier's shape, that can be shrink or widened or completely deformed. Basically, moving ρ we're just cutting the classifier predictor function at different levels, so different choices of ρ corresponds to different level sets of the classifier predictor. Although the role of ρ can remind a sort of radius, this interpretation is not totally true: ρ can be negative. But with a little abuse of formality, we can still maintain the parallelism as clearly reported in Figure 3.1.

3.2.1 Scalable Classifiers

As described in [170], a scalable classifier is a special classifier of this form

$$\phi_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) \doteq \begin{cases} +1 & \text{if } f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) < 0, \\ -1 & \text{otherwise.} \end{cases} \tag{3.4}$$

that satisfies the next **Assumption**:

Assumption 1 (Scalable Classifier)**.** Assume that for every $\boldsymbol{x} \in \mathcal{X}$, $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho)$ is a continuous and monotonically increasing function on ρ, i.e.

$$\rho_1 > \rho_2 \;\Rightarrow f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho_1) > f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho_2), \quad \forall \boldsymbol{x} \in \mathcal{X}. \tag{3.5}$$

Assume also that

$$\lim_{\rho \to -\infty} f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) < 0 < \lim_{\rho \to \infty} f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho), \quad \forall \boldsymbol{x} \in \mathcal{X}. \tag{3.6}$$

We sometimes will refer to the function $f_{\boldsymbol{\theta}} : \mathcal{X} \times \mathbb{R} \longrightarrow \mathbb{R}$ as *classifier predictor*. It is worth noting that $f_{\boldsymbol{\theta}}$ depends also on a second set of parameters $\boldsymbol{\theta} = \boldsymbol{\theta} = [\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{n_{\boldsymbol{\theta}}}]^\top \subset \mathbb{R}^{n_{\boldsymbol{\theta}}}$, the so-called *hyperparameters*, that is, all those parameters to be set in the model (e.g. different choices of kernel, regularization parameters, etc.). Obviously, a different choice of $\boldsymbol{\theta}$ corresponds to a possibly very different classifier. The role of different choices of $\boldsymbol{\theta}$ in the construction of the classifier is extremely important, and will be discussed in Section 3.3.

In the sequel, with some slight abuse of notation, we will sometimes refer to $f_{\boldsymbol{\theta}}$ as the classifier itself. The role of the parameter ρ plays a central role in the definition of the safety region, and this is an aspect that was missing with the "data-driven" definition of the safety region, as defined in the previous chapter. The first main difference is that this parameter can be tuned and not "learned" as it was training multiple SVDD to define a zero-false-positive zone. So, to understand properly how these new safety regions work, we have to focus more on the properties of these special classifiers.

Property 3.3. Suppose that Assumption 1 holds. Then, for each $\boldsymbol{x} \in \mathcal{X}$, there exists a unique $\bar{\rho}(\boldsymbol{x})$ satisfying $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \bar{\rho}(\boldsymbol{x})) = 0$. Moreover, the classifier $\phi_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho)$ given by (3.4) satisfies

$$\phi_\theta(\boldsymbol{x}, \rho) = -1 \Leftrightarrow \rho \geq \bar{\rho}(\boldsymbol{x}).$$

Proof. Because of (3.6) we have that if ρ is small enough then $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) < 0$. On the other hand, if ρ is large enough then $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) > 0$. This, along with the continuity nature of $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho)$, guarantees the existence of ρ such that $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = 0$. The uniqueness follows from the monotonic assumption on $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho)$. Denote $\bar{\rho}(\boldsymbol{x})$ this unique value of ρ satisfying $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \bar{\rho}(\boldsymbol{x})) = 0$. From the monotonically increasing nature of $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho)$ we have

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) \geq 0 \quad \Leftrightarrow \quad \rho \geq \bar{\rho}(\boldsymbol{x}).$$

Thus,

$$\phi_\theta(\boldsymbol{x}, \rho) = -1 \Leftrightarrow \rho \geq \bar{\rho}(\boldsymbol{x}).$$

□

Under Assumption 1, we denote $\bar{\rho}(\boldsymbol{x})$ the unique solution (see Property 3.3) to the equation

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = 0.$$

Strictly speaking, a scalable classifier can always be scaled such that the classification boundary passes through a chosen point: given $\boldsymbol{x}$, there is always a value of ρ, denoted $\bar{\rho}(\boldsymbol{x})$, that establishes the border between the two classes.
Therefore,another interpretation is that a SC is a classifier that maintains the target class of a given feature vector $\boldsymbol{x}$ under an increase of ρ. We also remark that this definition is implied by condition (3.5). Indeed, for a given $\tilde{\boldsymbol{x}} \in \mathcal{X}$ and $\rho_1 > \rho_2$, if $f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}, \rho_1) < 0$ (i.e. $\phi_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}, \rho_1) = +1$) then $f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}, \rho_2) < f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}, \rho_1) < 0$ (i.e. $\phi_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}, \rho_2) = +1$).

The next is a pivotal property that makes the difference with the previous approach: it shows that any standard binary classifier can be rendered scalable by simply including the scaling parameter ρ in an additive way. This is a remarkable result, since it means that it is possibile to define a safety region for *any* classifier and not only the SVDD like before.

Property 3.4. Consider the function $\hat{f}_{\boldsymbol{\theta}} : \mathcal{X} \to \mathbb{R}$ and its corresponding classifier

$$\hat{\phi}_{\boldsymbol{\theta}}(\boldsymbol{x}) \doteq \begin{cases} +1 & \text{if } \hat{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) < 0, \\ -1 & \text{otherwise.} \end{cases}$$

Then, the function $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = \hat{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \rho$ satisfies Assumption 1 and thus provides the scalable classifier

$$\phi_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) \doteq \begin{cases} +1 & \text{if } f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) < 0, \\ -1 & \text{otherwise.} \end{cases}$$

Proof. The result is trivial because $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = \hat{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \rho$ is clearly a continuous and monotonically increasing function on ρ. It is also straightforward to check that (3.6) is satisfied. □

The next example illustrates the use of the previous property to obtain a scalable classifier from a standard linear classifier.

Example 3.5 (Linear classifier as SC)**.** A simple example of a classifier that belongs to this class is the linear classifier

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} - b.$$

The classifier elements $\boldsymbol{w}, b$ may be obtained, for instance, as the solution of a SVM problem of the form

$$\min_{\boldsymbol{w},b} \frac{1}{2\eta} \boldsymbol{w}^\top \boldsymbol{w} + \frac{1}{2} \sum_{i=1}^{n} \max \left\{ 0, 1 - y_i(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - b) \right\}.$$

In this case, the classifier depends on the choice of the regularization term η, and of the specific regressor functional $\varphi(\cdot)$. That is, for a fixed choice of regressor, the hyperparameter vector is just the scalar $\boldsymbol{\theta} = \eta$. In this sense, we remark that a more rigorous notation would be $\boldsymbol{w} = \boldsymbol{w}(\boldsymbol{\theta})$ and $b = b(\boldsymbol{\theta})$, but we omit this dependence for the sake of readability.
It is immediate to observe that linear classifiers belong indeed to the class of scalable classifiers if we introduce a scaling parameter ρ, that is

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = \boldsymbol{w}^\top \boldsymbol{x} - b + \rho. \tag{3.7}$$

Indeed, given $\rho_1 > \rho_2$ we immediately have that

$$\boldsymbol{w}^\top \boldsymbol{x} - b + \rho_1 > \boldsymbol{w}^\top \boldsymbol{x} - b + \rho_2, \quad \forall \boldsymbol{x} \in \mathcal{X},$$

and it is straightforward to see that also (3.6) holds.

3.2.2 Probabilistic Safety Regions

Consider a given SC classifier $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho)$ (i.e. a classifier designed considering a specific choice of hyperparameter $\boldsymbol{\theta}$ controlled by a scalable parameter ρ). As previously anticipated, the parameter ρ can be seen as the height of the level set of the classifier predictor corresponding to ρ, that we will refer to as *ρ-safe set*:

$$\mathcal{S}(\rho) = \{\, \boldsymbol{x} \in \mathcal{X} \;:\; f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) < 0 \,\},$$

which represents the set of points $\boldsymbol{x} \in \mathcal{X}$ predicted as safe by the classifier with the specific choice ρ i.e. the safety region of the classifier $f_{\boldsymbol{\theta}}$ for given ρ. Of course, the larger ρ, the larger the region. Indeed, being $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho)$ a scalable classifier it is easy to see that

$$\rho_1 > \rho_2 \Longrightarrow \mathcal{S}(\rho_1) \supset \mathcal{S}(\rho_2).$$

This behavior is depicted in Fig. 3.1. In the next section, the reader can find some

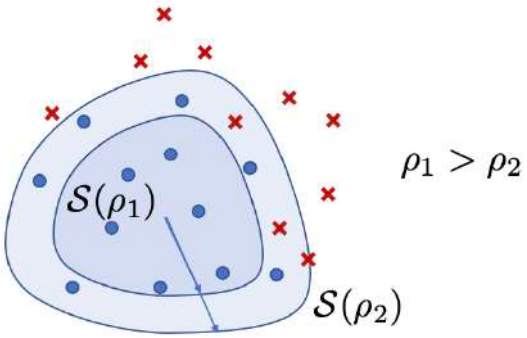


Figure 3.1: Graphical depiction of the role of the scaling parameter. The blue cirles represent safe points $\boldsymbol{x} \hat{=} \mathrm{S}$, while the red crosses represent unsafe ones, $\boldsymbol{x} \hat{=} \mathrm{U}$.

notable examples of well-assessed classifiers which can be reformulated in a way so that they belong to the SC family. The main result of the work consists in the definition of a level of probability to the safety region, which is possible starting from the definition of a calibration set $\mathcal{Z}_c \doteq \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c}$: a probabilistic safety region $\mathcal{S}_\varepsilon$, with a probability no smaller than $1 - \delta$, satisfies the probability constraint

$$\Pr\Big\{ y = -1 \text{ and } \boldsymbol{x} \in \mathcal{S}_\varepsilon \Big\} \leq \varepsilon.$$

We will assume that the pair $(\boldsymbol{x}, y)$ is a random variable and that $\Pr\{\boldsymbol{x} \in \mathcal{X}\} = 1$. Moreover, the n_c samples of $\mathcal{Z}_c$ are assumed i.i.d..

Theorem 3.6 (Probabilistic Safety Region)**.** *Consider the classifier* (3.4)*, and suppose that Assumption 1 holds and that* $\Pr\{\boldsymbol{x} \in \mathcal{X}\} = 1$*. Given a calibration set* $\mathcal{Z}_c \doteq \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c}$ *(n_c i.i.d. samples), suppose that* $\delta \in (0, 1)$*,* $\varepsilon \in (0, 1)$*, and the integer discarding parameter* r *satisfies* $n_c \geq r \geq 1$*, and*

$$\mathbf{B}(r - 1; n_c, \varepsilon) \leq \delta.$$

Consider the subset $\mathcal{Z}_c^U = \{(\tilde{\boldsymbol{x}}_j^U, -1)\}_{j=1}^{n_U}$ *corresponding to all the unsafe samples in* $\mathcal{Z}_c$ *and define the* probabilistic radius of level ε

$$\rho_\varepsilon \doteq \max^{(r)} \left(\{\bar{\rho}(\tilde{\boldsymbol{x}}_j^U)\}_{j=1}^{n_U} \right). \tag{3.8}$$

Then, define the set

$$\mathcal{S}_\varepsilon \doteq \begin{cases} \mathcal{S}(\rho_\varepsilon) & \text{if } n_U \geq r \\ \mathcal{X} & \text{otherwise.} \end{cases}$$

Then, with probability no smaller than $1-\delta$,

$$\Pr\Big\{ y = -1 \text{ and } \boldsymbol{x} \in \mathcal{S}_\varepsilon \Big\} \leq \varepsilon. \tag{3.9}$$

Proof. Let us introduce the auxiliary function

$$\tau : \mathcal{X} \times \{-1, 1\} \to [-1, 1),$$

which is defined as

$$\tau(\boldsymbol{x}, y) \doteq \begin{cases} -1 & \text{if } y = +1, \\ \frac{\bar{\rho}(\boldsymbol{x})}{1+|\bar{\rho}(\boldsymbol{x})|} & \text{otherwise.} \end{cases} \tag{3.10}$$

Denote now

$$\tau_\varepsilon = \max^{(r)} (\{\tau(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c})).$$

Since $\delta \in (0,1)$, $\varepsilon \in (0,1)$ and the integers $n_c \geq r \geq 1$ satisfy

$$\sum_{i=0}^{r-1} \binom{n_c}{i} \varepsilon^i (1-\varepsilon)^{n_c - i} \leq \delta,$$

we have from Property 3.1 that, with a probability no smaller than $1-\delta$,

$$\Pr\Big\{ \tau(\boldsymbol{x}, y) > \tau_\epsilon \Big\} \leq \varepsilon. \tag{3.11}$$

The rest of the proof shows that the previous inequality is equivalent to the claim of the theorem. That is,

$$\Pr\Big\{ \tau(\boldsymbol{x}, y) > \tau_\varepsilon \Big\} \leq \varepsilon \quad \Leftrightarrow \quad \Pr\Big\{ y = -1 \text{ and } \boldsymbol{x} \in S_\epsilon \Big\} \leq \varepsilon.$$

We consider two cases $n_U < r$ and $n_U \geq r$.

- **Case** $n_U < r$: By definition,

  $$-1 = \tau(\boldsymbol{x}, +1) < \tau(\boldsymbol{x}, -1) \in (-1, 1), \ \forall \boldsymbol{x} \in \mathcal{X}.$$

  This means that the smallest values for $\tau(\boldsymbol{x}, y)$ are attained at the safe samples. From $n_U < r$ we have that at most $r-1$ elements of the calibration set correspond to unsafe samples. Equivalently, no more than $r-1$ elements of

$\{\tau(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c}$ are larger than -1. This implies that the r-th largest value in $\{\tau(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c}$ corresponds to a safe sample and is equal to -1. That is,

$$\tau_\varepsilon = \max^{(r)}\left(\{\tau(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c}\right) = -1.$$

Thus, the inequality (3.11) is equivalent in this case to

$$\Pr\Big\{\tau(\boldsymbol{x}, y) > -1\Big\} \leq \varepsilon.$$

By definition, for every $\boldsymbol{x} \in \mathcal{X}$ we have

$$\tau(\boldsymbol{x}, y) > -1 \;\Leftrightarrow\; y = -1.$$

Thus, we obtain that in this case, $\tau_\varepsilon = -1$ and

$$\begin{aligned}\Pr\Big\{\tau(\boldsymbol{x}, y) > \tau_\varepsilon\Big\} \leq \varepsilon \quad &\Leftrightarrow \quad \Pr\Big\{y = -1\Big\} \leq \varepsilon \\ &\Leftrightarrow \quad \Pr\Big\{y = -1 \text{ and } \boldsymbol{x} \in \mathcal{X}\Big\} \leq \varepsilon.\end{aligned}$$

From the assumptions of the Theorem, we have that, by definition, $n_U < r$ implies $S_\varepsilon = \mathcal{X}$. Thus, we conclude that in this case,

$$\Pr\Big\{\tau(\boldsymbol{x}, y) > \tau_\varepsilon\Big\} \leq \varepsilon \;\Leftrightarrow \Pr\Big\{y = -1 \text{ and } \boldsymbol{x} \in S_\varepsilon\Big\} \leq \varepsilon.$$

- **Case** $n_U \geq r$: In this case, the r-largest value of $\{\tau(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c}$ is attained at an element of the unsafe calibration set $\mathcal{Z}_c^U = \left\{(\tilde{\boldsymbol{x}}_j^U, -1)\right\}_{j=1}^{n_U} \subseteq \mathcal{Z}_c$. That is,

$$\begin{aligned}\tau_\varepsilon &= \max^{(r)}\left(\{\tau(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c}\right) \\ &= \max^{(r)}(\{\tau(\tilde{\boldsymbol{x}}_j^U, -1)\}_{j=1}^{n_U}) \in (-1, 1).\end{aligned}$$

Define now

$$\rho_\varepsilon = \max^{(r)}\left(\{\bar{\rho}(\tilde{\boldsymbol{x}}_j^U)\}_{j=1}^{n_U}\right).$$

Since $\frac{\bar{\rho}(\boldsymbol{x})}{1+|\bar{\rho}(\boldsymbol{x})|}$ is a monotonically increasing function on $\bar{\rho}(\boldsymbol{x})$, we have that τ_ϵ can be obtained by means of ρ_ε. That is,

$$\tau_\varepsilon = \max^{(r)}(\{\tau(\tilde{\boldsymbol{x}}_j^U, -1)\}_{j=1}^{n_U}) = \frac{\rho_\varepsilon}{1 + |\rho_\varepsilon|}.$$

Thus, from $\tau_\varepsilon > -1$ and the previous expression we obtain the equivalences

$$\begin{aligned}\tau(\boldsymbol{x}, y) > \tau_\varepsilon \quad &\Leftrightarrow \quad y = -1 \text{ and } \frac{\bar{\rho}(\boldsymbol{x})}{1 + |\bar{\rho}(\boldsymbol{x})|} > \frac{\rho_\varepsilon}{1 + |\rho_\varepsilon|} \\ &\Leftrightarrow \quad y = -1 \text{ and } \bar{\rho}(\boldsymbol{x}) \geq \rho_\varepsilon.\end{aligned}$$

Therefore, $\Pr\Big\{\tau(\boldsymbol{x}, y) > \tau_\varepsilon\Big\} \leq \varepsilon$ is equivalent to

$$\Pr\Big\{y = -1 \text{ and } \bar{\rho}(\boldsymbol{x}) > \rho_\varepsilon\Big\} \leq \varepsilon.$$

From the monotonicity of $f_\theta(\boldsymbol{x}, \rho)$ on ρ (Assumption 1) we obtain that the previous inequality can be rewritten as

$$\Pr\Big\{ y = -1 \text{ and } f_\theta(\boldsymbol{x}, \bar{\rho}(\boldsymbol{x})) > f_\theta(\boldsymbol{x}, \rho_\varepsilon) \Big\} \leq \varepsilon.$$

Taking into consideration that $f_\theta(\boldsymbol{x}, \bar{\rho}(\boldsymbol{x})) = 0$ we obtain that $\Pr\Big\{ \tau(\boldsymbol{x}, y) > \tau_\varepsilon \Big\} \leq \varepsilon$ is equivalent to

$$\Pr\Big\{ y = -1 \text{ and } f_\theta(\boldsymbol{x}, \rho_\varepsilon) < 0 \Big\} \leq \varepsilon.$$

From the assumptions of the Theorem we have that, by definition, $n_U \geq r$ implies that S_ε is equal to $\{ \boldsymbol{x} \in \mathcal{X} \ : \ f_\theta(\boldsymbol{x}, \rho_\varepsilon) < 0 \}$. Thus, we conclude in this case that

$$\Pr\Big\{ \tau(\boldsymbol{x}, y) > \tau_\varepsilon \Big\} \leq \varepsilon \ \Leftrightarrow \Pr\Big\{ y = -1 \text{ and } \boldsymbol{x} \in S_\varepsilon \Big\} \leq \varepsilon.$$

□

3.2.2.1 eXample of Scalable Classifiers

We consider three state-of-the-art classifiers which possess the "scalability" property: Support Vector Machines, Support Vector Data Description and Logistic Regression. In the following examples, we assume we are given a learning set

$$\mathcal{Z}_L \doteq \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subseteq \mathcal{X} \times \{-1, +1\}$$

containing observed feature points and corresponding labels $\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)$. Then, we introduce the kernels (see e.g. [29]). In particular, letting

$$\varphi : \mathcal{X} \longrightarrow \mathcal{V}$$

be a *feature map* (where $\mathcal{V}$ is an inner product space) we define

$$\Phi = \begin{bmatrix} \varphi(\boldsymbol{x}_1) & \varphi(\boldsymbol{x}_2) & \dots & \varphi(\boldsymbol{x}_n) \end{bmatrix}, \tag{3.12}$$

$$D = \operatorname{diag}\{y_1, y_2, \dots, y_n\}, \tag{3.13}$$

$$K = \Phi^\top \Phi, \tag{3.14}$$

with $K_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_j),\ i \in [n], j \in [n]$ the kernel matrix.

To get an accurate derivation of the models, the reader can consult Section 3.5.

The models considered and their derivation is absolutely classical. However, since we are interested in scalable classifiers with guaranteed safety, for each model we will consider two hyperparameters, i.e. we will set $\boldsymbol{\theta} = [\eta, \tau]^\top$, where besides the classical regularization parameter $\eta \in \mathbb{R}$ we introduce a weighting term $\tau \in (0, 1)$ that penalizes missclassification errors (the role of τ is much in the spirit of quantile regression formulation [21]).

3.2.2.2 Scalable SVM

SVM is the simplest extension of a linear model and indeed we define its classifier predictor as

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{w}^\top \varphi(\boldsymbol{x}) - b.$$

The SVM formulation we adopt is the classical one proposed by Vapnik in [5], with the addition of the weighting parameter τ:

$$\begin{aligned} \min_{\boldsymbol{w},b,\xi_1,\ldots,\xi_n} \quad & \frac{1}{2\eta}\boldsymbol{w}^\top \boldsymbol{w} + \frac{1}{2}\sum_{i=1}^{n} \left((1-2\tau)y_i + 1\right)\xi_i \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - b) \leq \xi_i - 1,\ i \in [n], \\ & \xi_i \geq 0,\ i \in [n]. \end{aligned} \tag{3.15}$$

The offset b can be found exploiting special feature points $\boldsymbol{x}_s$ called *support vectors* that are such that $\varphi(\boldsymbol{x}_s)$ lies on the boundary of the transformed space. The addition of the scaling parameter ρ changes the model in

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) \quad = \quad \boldsymbol{w}^\top \varphi(\boldsymbol{x}) - b + \rho.$$

For the linear kernel, the variation of ρ is simply a rigid translation of the classification hyperplane; for other kernels, for example the Gaussian kernel or the polynomial kernel, the effect is the "deflation" or the "inflation" of the classification boundary. The composition with the feature map does not affect the scalability property of the linear classifier, so it is easy to verify from the considerations made in 3.5 that indeed scalable SVM satisfies Assumption 1.
The reader can find a more in-depth description in Appendix 3.5.1.

Remark 3.7 (On the role of τ parameter.)**.** Indeed, it is easy to see that small values of τ adds more weight to the class $+1$, which is the class we are interested in. So, the choice of a "good" value of τ is particularly important. This will be discussed in Section 3.3, where the possibility of considering several values for this parameter in the context of our approach is discussed in detail.

3.2.2.3 Scalable SVDD

SVDD was introduced in [20] based on the idea of classifying the feature vectors by enclosing the target points (in the kernel space) in the smallest hypersphere of radius R and center $\boldsymbol{w}$. With this idea, we define the scalable classifier predictor for SVDD as

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = \|\varphi(\boldsymbol{x}) - \boldsymbol{w}\|^2 - (R^2 - \rho),$$

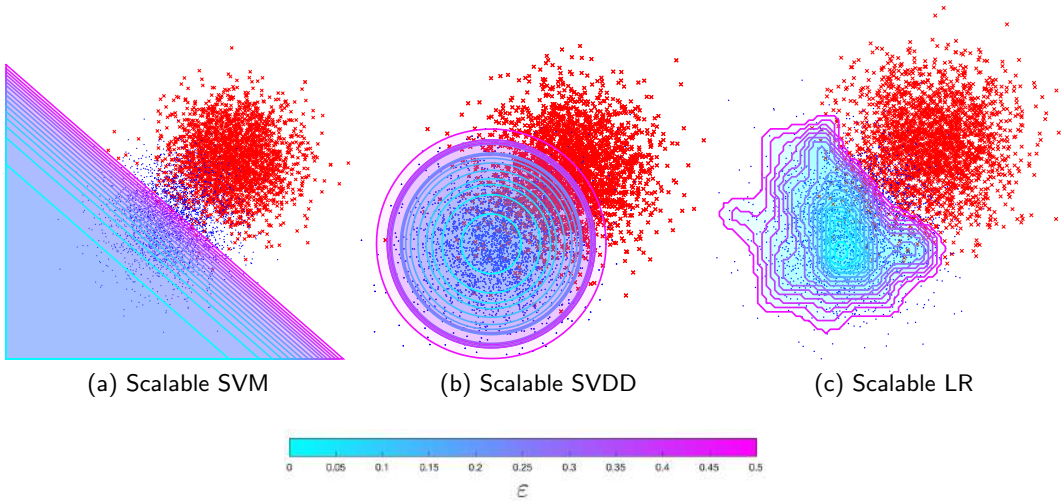


Figure 3.2: 2D examples of PSRs via, respectively from left to right, scalable SVM (linear kernel), scalable SVDD (linear kernel) and scalable LR (Gaussian kernel). Synthetic test data were sampled from Gaussian distributions and classified for varying values of ε (from 0.01 to 0.5, from lighter to darker colors), after calibrating the scalable parameters with a calibration set of size n_c according with bound (3.2) and $\delta = 10^{-6}$. Blue points refer to the safe class ($\boldsymbol{x} \hat{=} \mathrm{S}$) an reds to the unsafe one ($\boldsymbol{x} \hat{=} \mathrm{U}$).

where $\boldsymbol{w}, R$ are obtained as the solution of the following weighted optimization problem

$$\begin{aligned} \min_{\boldsymbol{w},R,\xi_1,\ldots,\xi_n} \quad & \frac{1}{2\eta}R^2 + \frac{1}{2}\sum_{i=1}^{n}\left((1-2\tau)y_i + 1\right)\xi_i \\ \text{s.t.} \quad & y_i\left(\|\varphi(\boldsymbol{x}_i) - \boldsymbol{w}\|^2 - R^2\right) \leq \xi_i,\ i \in [n], \\ & \xi_i \geq 0,\ i \in [n] \end{aligned} \tag{3.16}$$

that, again, depends on the hyperparameters $\boldsymbol{\theta} = [\eta, \tau]^\top$, playing the role of regularization and missclassification parameters. As for the SVM model, the radius R is retrieved by support vectors, that are feature points lying on the hypersphere boundary of the classification in the kernel space. It is immediate to observe that the introduction of the scaling parameter ρ maintains the idea of radius defining a new radius $\tilde{R} = \sqrt{R^2 - \rho}$, underlying then the meaning of scaling parameter. Indeed, the obtained scalable SVDD-classifier predictor is

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = \|\varphi(\boldsymbol{x}) - \boldsymbol{w}\|^2 - (R^2 - \rho),$$

which clearly satisfies equations (3.5) and (3.6) and then it belongs to the SC family. SVDD example is particularly significant in understanding the usefulness of a scalable classifier: in this case, the parameter to be scaled is effectively a radius. The

smaller the radius, the more conservative the prediction about the target class. Thus, the idea behind these classifiers is to have control over the prediction simply by handling a scalar value, enlarging or decreasing it as a radius.
The reader can find a more in-depth description in Appendix 3.5.2.

3.2.2.4 Scalable Logistic Regression

Logistic Regression (LR) classifies points $\boldsymbol{x} \in \mathcal{X}$ on the basis of the probability expressed by the logistic function

$$\begin{aligned} \frac{1}{1+e^{-\left(\boldsymbol{w}^\top \varphi(\boldsymbol{x})-b\right)}} &= \mathbb{P}\left\{y=+1 \mid \boldsymbol{x}\right\} \\ &= 1-\mathbb{P}\left\{y=-1 \mid \boldsymbol{x}\right\}, \end{aligned}$$

where $\boldsymbol{w}$ and b minimize the regularized negative log-likelihood

$$\begin{aligned} &\mathrm{L}\left(\boldsymbol{w}, b \mid \boldsymbol{x}, y\right) = \frac{1}{2\eta}\boldsymbol{w}^\top \boldsymbol{w} \\ &+\frac{1}{2}\sum_{i=1}^{n}\left((1-2\tau)y_i+1\right)\log\left(1+e^{-y_i\left(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i)-b\right)}\right), \end{aligned} \tag{3.17}$$

with b explicitly computed with the support vectors of the model.
Note that, differently from classical LR and in the spirit of previously described approaches, we introduced into L the weight parameter $\tau \in (0,1)$ to penalize misclassification. As before, the scalable classifier predictor for the LR can be defined as

$$\begin{aligned} f_{\boldsymbol{\theta}}(\boldsymbol{x},\rho) &= \frac{1}{2}-\frac{1}{1+e^{-\left(\boldsymbol{w}^\top \varphi(\boldsymbol{x})-b\right)+\rho}} \\ &= \frac{1}{2}\frac{e^{-\left(\boldsymbol{w}^\top \varphi(\boldsymbol{x})-b\right)+\rho}-1}{1+e^{-\left(\boldsymbol{w}^\top \varphi(\boldsymbol{x})-b\right)+\rho}}. \end{aligned}$$

The interested reader can find a more in-depth description in Appendix 3.5.3.
It is easy to show that LR is a scalable classifier with such a choice for ρ. To this end, it is sufficient to note that $f_\theta(\boldsymbol{x},\rho)$ is strictly increasing in ρ

$$\frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x},\rho)}{\partial \rho} = \frac{e^{-\left(\boldsymbol{w}^\top \varphi(\boldsymbol{x})-b\right)+\rho}}{\left(1+e^{-\left(\boldsymbol{w}^\top \varphi(\boldsymbol{x})-b\right)+\rho}\right)^2} > 0,$$

and that

$$\lim_{\rho\to-\infty} f_\theta(\boldsymbol{x},\rho) = -1/2 < 0 < 1/2 = \lim_{\rho\to+\infty} f_\theta(\boldsymbol{x},\rho).$$

That shows that LR is indeed a scalable classifier, since Assumption 1 holds.

Remark 3.8 (Generality of SC)**.** We remark that the three examples above, although already significant in themselves, represent only a small subset of possible scalable classifiers. Indeed, it is believed that the scalable classifier has been defined in rather general terms, and that a scalable formulation of "classical classifiers" may be derived, e.g. for a specific class of neural networks. For example it is easy to derive a scalable version of Perceptron [2]. However, we prefer not to dwell further on this and leave it to the reader to demonstrate that other classifiers are scalable.

Remark 3.9. We emphasize that one of the main advantages of our approach is that the distribution of the calibration set need not be equal to that of the learning set. It should be equal to the one for which we want to impose probabilistic guarantees. This is a crucial observation, since probabilistic guarantees apply only to the distribution from which the calibration set was drawn, which must therefore be chosen carefully. Note also that as the desired degree of guarantee changes, the cardinality required for the the calibration set changes.

Example 3.10. To give the reader a simple and meaningful idea of the method, Figure 3.2 shows the behavior of the PSR as ε varies while δ is fixed to 10^{-6}. For this example, we sampled with equal probability two classes, "safe" S and "unsafe" U, from two Gaussian distributions with respectively means and covariance matrices

$$\mu_{\mathrm{S}} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \Sigma_{\mathrm{S}} = \mathrm{I} \; ; \; \mu_{\mathrm{U}} = \begin{bmatrix} +1 \\ +1 \end{bmatrix}, \Sigma_{\mathrm{U}} = \mathrm{I}$$

where I is the identity matrix. We sampled $3,000$ points for the training set and $10,000$ for the test set, and $n_c = n_c(\varepsilon)$ points for the calibration set according to Corollary 3.2.
The behaviour of the PSR constructed via the scalable classifiers is in agreement with the theory developed: the smaller the ε (i.e. the smaller is the error required) the smaller is the PSR, to guarantee more probability of safety. For scalable SVM (left) and scalable SVDD (middle) we choose a linear kernel, while for scalable LR (right) a Gaussian kernel was used. The blue cirles represent safe points $\boldsymbol{x} \hat{=} \mathrm{S}$, while the red crosses represent unsafe ones, $\boldsymbol{x} \hat{=} \mathrm{U}$.

3.3 Finite families of hyperparameters

Probabilistic scaling guarantees confidence in prediction for any given scalable classifier. In other words, for any fixed value of hyperparameter $\boldsymbol{\theta}$, the safety set obtained selecting the scaling parameter ρ according to our procedure will fulfill the required probabilistic guarantees. However, different values of $\boldsymbol{\theta}$ will correspond to different models, and the resulting set will consequently be different, both in "size" and in "goodness". In particular, if the starting SC has been chosen badly, our procedure would lead to a very small PSR, that would be indeed guaranteed theoretically, but with no practical use.

Hence, the problem of selecting the best initial SC becomes of great importance. In our setup, this problem translates in choosing the best value for the hyperparameter. Also, we remark that, in general, there may be other parameters that affect the performance of a classifier, such as the choice of different kernels or different weights or different regularizations and many others. Hence, in general, the hyperparameter $\boldsymbol{\theta}$ may be of larger dimensions and consider several possible choices. For instance, the hyperpameter $\boldsymbol{\theta}$ may collect different values of the parameter τ used to correctly balance the classifier (see Remark 3.7), or different values of the regularization parameter η, or even specific choices of different kernels.

To formally state our problem, we assume to have a finite set of m possible hyperparameters to choose from

$$\Theta = \left\{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots, \boldsymbol{\theta}^{(m)}\right\}, \tag{3.18}$$

and we consider the problem of selecting the "best" one.
Hence, we assume we are given a performance function $J : \Theta \to \mathbb{R}$ which measures the goodness of the model described by $\boldsymbol{\theta}$. Then, we will choose

$$\boldsymbol{\theta}^{\star} \doteq \arg\max_{\boldsymbol{\theta}\in\Theta} J(\boldsymbol{\theta}).$$

Clearly, depending on the problem at end, different cost functions may be devised. We discuss a possible meaningful choice of performance function in Section 3.3.2. In the following section, we show how the scaling procedure can be easily modified to guarantee that the selected SC, and the ensuing estimate of the PSR, still enjoy the desired probabilistic guarantees.

3.3.1 Probabilistic scaling for finite families of SC

The following results, whose proof is a direct consequence of Bonferroni's inequality and is omitted for brevity, shows how the results in Theorem 3.6 may be immediately extended to the case of a finite family of classifiers (i.e. a finite set of candidate SCs described by a finite set of possible values of hyperparameters.

Theorem 3.11 (Probabilistic Safety Region for finite families of hyperparameters)**.** *Consider the classifier* (3.4)*, a finite set of possible hyperparameter values* $\boldsymbol{\theta} \in \Theta = \left\{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \dots, \boldsymbol{\theta}^{(m)}\right\}$*, and suppose that Assumption 1 holds and that* $\Pr\{\boldsymbol{x} \in \mathcal{X}\} = 1$*. Fix a risk parameter* $\varepsilon \in (0,1)$*, a probability level* $\delta \in (0,1)$ *and an integer discarding parameter* $r \geq 1$*. Given* $\mathcal{Z}_c^U = \left\{(\tilde{\boldsymbol{x}}_j^U, -1)\right\}_{j=1}^{n_U}$ *corresponding to all the unsafe samples in a calibration set* $\mathcal{Z}_c$ *of* $n_c \geq r$ *i.i.d. samples, for all* $\boldsymbol{\theta}^{(k)}$*,* $k \in [m]$*, compute the corresponding scaling factors:*

- *compute the scaling parameters*

$$\bar{\rho}_j^{(k)} \text{ such that } f_{\boldsymbol{\theta}^{(k)}}(\tilde{\boldsymbol{x}}_j^U, \bar{\rho}_j^{(k)}) = 0, \quad j \in [n_U], k \in [m],$$

- *compute the k-th* probabilistic radius *and the k-th* probabilistic safety region of level ε, *i.e.*

$$\rho_{\varepsilon}^{(k)} \doteq \max^{(r)}\left(\{\bar{\rho}_j^{(k)}\}_{j=1}^{n^U}\right), \tag{3.19}$$

$$\mathcal{S}_{\varepsilon}^{(k)} \doteq \begin{cases} \mathcal{S}\left(\rho_{\varepsilon}^{(k)}\right) & \text{if } n_U \geq r \\ \mathcal{X} & \text{otherwise.} \end{cases}. \tag{3.20}$$

Then, the following holds

$$\Pr\left\{\Pr\left\{y = -1 \text{ and } \boldsymbol{x} \in \mathcal{S}_{\varepsilon}^{(k)}\right\} \leq \varepsilon\right\} \geq 1 - m\mathbf{B}(r-1; n_c, \varepsilon), \tag{3.21}$$

$\forall k \in [m]$.

In particular, this means that all sets $\mathcal{S}_{\varepsilon}^{(k)}$ are valid PSR candidates, and we have the possibility of selecting among those the "best" one according to some specific measure on how we expect the SC to behave. In the next subsection, we propose a possible criterion which proved to be very effective in our experience.

3.3.2 Increase of safe points

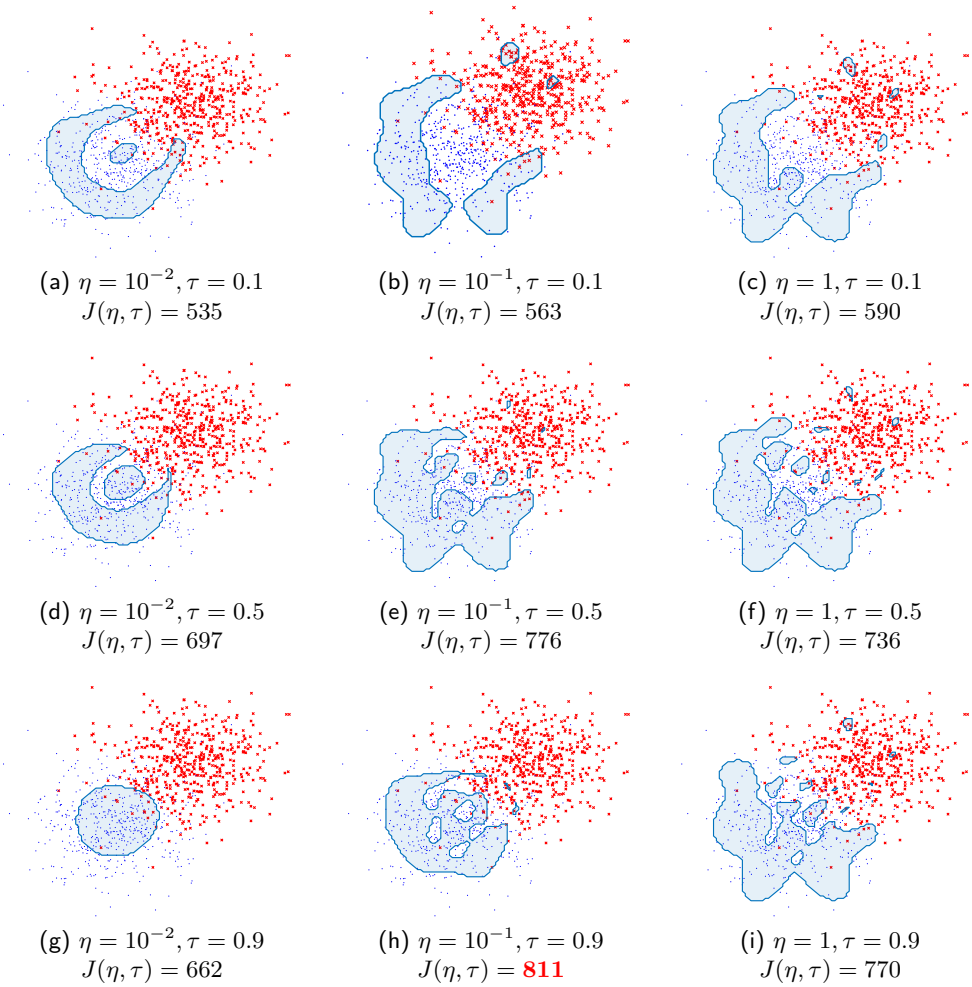
In general, one is interested in a solution which, besides providing probabilistic guarantees on the safe region, i.e. minimizing the probability of having unsafe points in the set $\mathcal{S}_{\varepsilon}^{(k)}$, it also maximises the number of safe points captured by the region itself. To this end, we first notice that, when applying the scaling procedure, we are basically only exploiting the unsafe points in calibration set $\mathcal{Z}_c$ (i.e. the points belonging to $\mathcal{Z}_c^U$).
It is thus immediately to observe that the remaining points in the calibration set, i.e. the points belonging to

$$\mathcal{Z}_c^S = \mathcal{Z}_c \setminus \mathcal{Z}_c^U,$$

i.e. the set containing all the safe (+1) points in $\mathcal{Z}_c$ may be exploited in evaluating the goodness of the candidate sets. To this end, given a candidate set $\mathcal{S}_{\varepsilon}^{(k)}$, we can measure its goodness as and select as performance function the cardinality of such set

$$J(\boldsymbol{\theta}^{(k)}) \doteq \left|\left\{\boldsymbol{z}_i \in \mathcal{Z}_c^S : \boldsymbol{z}_i \in \mathcal{S}_{\varepsilon}^{(k)}\right\}\right|. \tag{3.22}$$

Example 3.12. Considering the scalable SVDD with Gaussian kernel, in the same design as Example3.10, but with a probability of sampling outliers per class set at $p_O = 0.1$ (to allow for some noise), with only $1,000$ points for the test set (to make the boundary plot clearer, see Figure 3.3) and with ε set to 0.05 (that gives a calibration set with $2,064$ points), we computed the probabilistic safety region $\mathcal{S}_{\varepsilon}$ for different values of the hyperparameters $\boldsymbol{\theta} = [\eta, \tau]$, specifically $\eta = [10^{-2}, 10^{-1}, 1]$ and $\tau = [0.1, 0.2, \ldots, 0.9]$. All regions satisfy the probabilistic bound on the number of unsafe points within $\mathcal{S}_{\varepsilon}$, i.e. $\Pr\{\boldsymbol{x} \hat{=} \text{U} \text{ and } \boldsymbol{x} \in \mathcal{S}_{\varepsilon}\} < 0.05$, but the area covered



(a) $\eta = 10^{-2}, \tau = 0.1$ $J(\eta, \tau) = 535$

(b) $\eta = 10^{-1}, \tau = 0.1$ $J(\eta, \tau) = 563$

(c) $\eta = 1, \tau = 0.1$ $J(\eta, \tau) = 590$

(d) $\eta = 10^{-2}, \tau = 0.5$ $J(\eta, \tau) = 697$

(e) $\eta = 10^{-1}, \tau = 0.5$ $J(\eta, \tau) = 776$

(f) $\eta = 1, \tau = 0.5$ $J(\eta, \tau) = 736$

(g) $\eta = 10^{-2}, \tau = 0.9$ $J(\eta, \tau) = 662$

(h) $\eta = 10^{-1}, \tau = 0.9$ $J(\eta, \tau) = \mathbf{811}$

(i) $\eta = 1, \tau = 0.9$ $J(\eta, \tau) = 770$

Figure 3.3: Plots of PSRs at the $\varepsilon = 0.05$ level for Gaussian SVDD with different regularization parameters (η) and different weights (τ). The shape of the region changes by varying the design parameters, but maintaining the probabilistic guarantee on the number of unsafe points within it. The best configuration is chosen by maximizing a performance index, in this case the number of calibration points contained in the region (see the equation (3.22)). For this toy example, the best configuration is obtained for $\epsilon = 10^{-1}$ and $\tau = 0.9$, but others can be found by increasing the number of candidate design parameters.

changes as the design parameters change. The best region can be chosen as the one that maximizes an index parameter, as in this case the equation (3.22) that increases the number of safe points in the PSR.

Finally, it is worth noting that the parameter to be optimized can be specified in principle according to the specific problem to be solved. For example, J can

be defined such that it maximizes accuracy or minimizes only FPR or FNR or maximizes AUC and so on.

Table 3.1: Table of the performance of PSRs for vehicle platoon as the collision probability ($\varepsilon = 0.01, 0.05, 0.1$) and design parameters ($\eta = 10^{-2}, 10^{-1}, 1$ and $\tau = 0.1, 0.5, 0.9$) vary. In red, the best results for each combination of classifier and parameters.

		$\eta = 10^{-2}$						$\eta = 10^{-1}$						$\eta = 1$					
		$\tau = 0.1$		$\tau = 0.5$		$\tau = 0.9$		$\tau = 0.1$		$\tau = 0.5$		$\tau = 0.9$		$\tau = 0.1$		$\tau = 0.5$		$\tau = 0.9$	
		Pr{}	$J(\boldsymbol{\theta})$	Pr{}	$J(\boldsymbol{\theta})$	Pr{}	$J(\boldsymbol{\theta})$	Pr{}	$J(\boldsymbol{\theta})$	Pr{}	$J(\boldsymbol{\theta})$	Pr{}	$J(\boldsymbol{\theta})$	Pr{}	$J(\boldsymbol{\theta})$	Pr{}	$J(\boldsymbol{\theta})$	Pr{}	$J(\boldsymbol{\theta})$
		$\varepsilon = 0.01$																	
Vehicle Platooning	SC-SVM	0.04	825	0.02	1254	0.03	1054	0.02	1211	0.01	1839	0.01	2030	0.01	1390	0.01	1833	**0.01**	**2361**
	SC-SVDD	0.01	3347	0.02	2395	0.01	3532	0.01	3342	0.02	2395	0.01	3547	0.01	3342	0.01	3032	**0.01**	**3548**
	SC-LR	0.02	5373	0.02	5372	0.02	5372	**0.02**	**5378**	0.02	5373	0.02	5372	0.02	5372	0.02	5376	0.02	5350
		$\varepsilon = 0.05$																	
	SC-SVM	0.06	486	0.07	568	0.08	518	0.05	707	0.05	780	0.06	752	0.05	805	0.04	797	**0.05**	**844**
	SC-SVDD	0.05	886	0.06	678	0.05	876	**0.05**	**889**	0.06	678	0.05	880	**0.05**	**889**	0.06	763	0.05	880
	SC-LR	0.03	909	0.03	910	0.03	909	0.03	915	0.03	911	0.02	907	**0.03**	**953**	0.03	951	0.00	889
		$\varepsilon = 0.1$																	
	SC-SVM	0.10	360	0.12	394	0.16	357	0.14	430	0.10	456	0.10	449	0.12	466	0.09	495	**0.09**	**487**
	SC-SVDD	0.10	508	0.11	463	0.09	528	0.10	508	0.11	463	**0.09**	**529**	0.09	508	0.10	485	**0.09**	**529**
	SC-LR	0.10	566	0.10	574	0.10	574	0.10	577	0.09	586	0.09	567	**0.09**	**597**	0.07	568	0.06	558

3.4 A real-world application: Vehicle Platooning

Safety critical assessment is highly required in the automotive industry and vehicle platooning (VP) [62] represents one of the most challenging CPS (Cyber Physical System) in this context. The main goal of VP is to find the best trade-off between performance (i.e., maximizing speed and minimizing vehicle mutual distance) and safety (i.e., collision avoidance). With the idea of finding the largest region in the input space where safety is probabilistically guaranteed, we tested our scalable classifiers on the following scenario: given the platoon at a steady state of speed and reciprocal distance of the vehicles, a braking is applied by the leader of the platoon [50, 72]. Safety is referred to a collision between adjacent vehicles (in the study, it is actually registered when the reciprocal distance between vehicles achieves a lower bound, e.g. 2 m) and the dynamic of the system is generated by the following differential equations [50]:

$$\begin{cases} \dot{v} = \frac{1}{m_i}(F_i - (a_i + b_i \cdot v_i^2)) \\ \dot{d}_i = v_{i-1} - v_i \end{cases} \tag{3.23}$$

where v_i, m_i, a_i, b_i and F_i are, respectively, the speed, the mass, the tire-road rolling distance, the aerodynamic drag and the braking force (the control law) of vehicle i and d_i is the distance of vehicle i from the previous one $i-1$.
The behaviour of the dynamical system is synthesised by the following vector of features:

$$\mathbf{I} = [N, \boldsymbol{\iota}(0), F_0, \mathbf{m}, \mathbf{q}, \mathbf{p}] \tag{3.24}$$

$N+1$ being the number of vehicles in the platoon, $\boldsymbol{\iota} = [\mathbf{d}, \mathbf{v}, \mathbf{a}]$ are the vectors of reciprocal distance, speed, and acceleration of the vehicles, respectively ($\boldsymbol{\iota}(0)$ denotes that the quantities are sampled at time $t = 0$, after which a braking force is applied by the leader [72] and simulations are set in order to manage possible transient periods and achieve a steady state of $\boldsymbol{\iota}$ before applying the braking.), $\mathbf{m}$ is the vector of weights of the vehicles, F_0 is the braking force applied by the leader, $\mathbf{q}$ is the vector of quality measures of the communication medium (fixed delay and packet error rate (PER) are considered in the simulations) and finally $\mathbf{p}$ is the vector of tuning parameters of the control scheme.

The Plexe simulator [72, 50] has been used to register $20,000$ observations in the following ranges: $N \in [3, 8]$, $F_0 \in [-8, -1] \times 10^3 N$, $\mathbf{q} \in [0, 0.5]$, $\mathbf{d}(0) \in [4, 9]$ m, $\mathbf{v}(0) \in [10, 90]$ Km/h. Initial acceleration $\mathbf{a}(0)$ is computed as $\mathbf{a}(0) = F_0/\mathbf{m}$ Km/h^2.

In the same setting of, we searched safety for three levels of guarantee ($\varepsilon = 0.01, 0.05, 0.1$) and different hyperparameters ($\eta = 10^{-2}, 10^{-1}, 1$ and $\tau = 0.1, 0.5, 0.9$), evaluating the performance (reported in Table 3.3.2) computing the probability of getting a collision inside the "non-collision" probabilistic safety region $\mathcal{S}_\varepsilon$, $\Pr\Big\{\mathbf{x} \hat{=} \{\text{collision}\} \text{ and } \boldsymbol{x} \in \mathcal{S}_\varepsilon\Big\}$, and the number of non-collision points of the calibration set contained in $\mathcal{S}_\varepsilon$, varying the hyperparameters, $J(\boldsymbol{\eta}, \boldsymbol{\tau})$. We divided the dataset in training set ($n_{tr} = 3,000$ points), calibration set ($n_c = 10,320, 2,064, 1,032$ respectively for $\varepsilon = 0.01, 0.05, 0.1$) and test set ($n_{ts} = n_{tr} - n_c$).

The results obtained are very satisfactory, considering that the dataset is very noisy. The error level is limited as expected by ε, despite some uncertainty due to the complexity of the classification. For all scalable classifiers, the trade-off between the guarantee and the number of safe points of the calibration set within the "non-collision" safety region is good, allowing for the construction of operational regions where safety can be guaranteed. In particular, the best performance obtained by each classifier at different levels of ε is highlighted in red. Furthermore, Figure 3.4 shows the trend of the probability of getting a collision within the safety region as ε varies, with $\eta = 1$ and $\tau = 0.5$ (i.e., without regularizing and weighting equally both the classes). As expected, the behavior is (almost) linear with ε, with SC-LR deviating slightly from SC-SVM and SC-SVDD.

3.5 Small appendix for Scalable Classifiers

3.5.1 Scalable SVM

Proof. Operationally, it is better to work with the dual form of the SVM formulation in (3.15) . Defining the multipliers

$$\boldsymbol{\alpha} = [\alpha_1 \, \alpha_2 \, \cdots \, \alpha_n]^\top \tag{3.25}$$

$$\boldsymbol{\beta} = [\beta_1 \, \beta_2 \, \cdots \, \beta_n]^\top \tag{3.26}$$

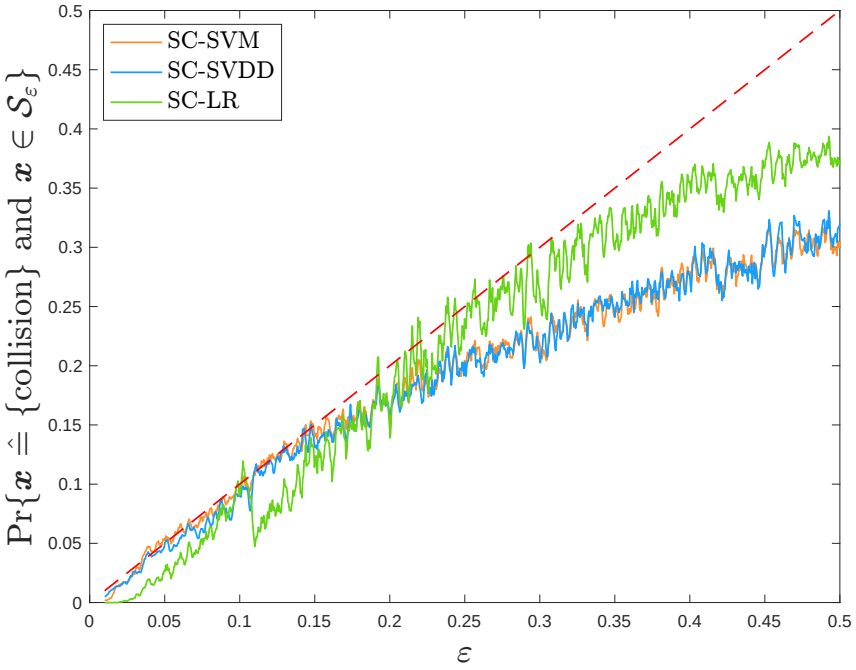


Figure 3.4: Probability of getting a collision as ε varies. The trend is (almost) linear for all the classifiers.

the Lagrangian is given by

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) =& \frac{1}{2\eta}\boldsymbol{w}^\top \boldsymbol{w} + \frac{1}{2}\sum_{i=1}^{n}\left((1-2\tau)y_i + 1\right) \\ &- \sum_{i=1}^{n}\alpha_i\left[y_i\left(b - \boldsymbol{w}^\top\varphi(\boldsymbol{x}_i)\right) - 1 + \xi_i\right] - \sum_{i=1}^{n}\beta_i\xi_i.\end{aligned} \tag{3.27}$$

Setting partial derivatives to zero, we obtain the following constraints:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = -\eta\sum_{i=1}^{n}\alpha_i y_i\varphi(\boldsymbol{x}_i) \tag{3.28}$$

$$\frac{\partial\mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0, \tag{3.29}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\xi_i} = 0 \Rightarrow \beta_i &= -\alpha_i + \frac{1}{2}\left((1-2\tau)\,y_i + 1\right) \\ &\Rightarrow 0 \leq \alpha_i \leq \frac{1}{2}\left((1-2\tau)\,y_i + 1\right)\; i \in [n].\end{aligned} \tag{3.30}$$

Substituting (3.28), (3.29) and (3.30) into (3.27) we obtain

$$\mathcal{L} = -\frac{\eta}{2}\sum_{i=1}^{n}\alpha_i\alpha_j y_i y_j\varphi(\boldsymbol{x}_i)^\top\varphi(\boldsymbol{x}_j),$$

and using the notation introduced in (7.13), (7.14) and (7.15) we can write $\mathcal{L}$ in a more compact form as $\mathcal{L} = -\frac{\eta}{2}\boldsymbol{\alpha}^\top DKD\boldsymbol{\alpha}\sum_{i=1}^n + \boldsymbol{\alpha}$, that has to be maximized with respect the dual variables α_i, $i \in [n]$ and (3.35),(3.36) constraints.
Then the dual formulation of the scalable SVM is

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{\eta}{2}\boldsymbol{\alpha}^\top DKD\boldsymbol{\alpha} + \boldsymbol{\alpha} \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0, \\ & 0 \le \alpha_i \le \frac{1}{2}\left((1-2\tau)\, y_i + 1\right),\ i \in [n], \end{aligned} \tag{3.31}$$

and the vector of weights is obtained as

$$\boldsymbol{w} = -\eta \sum_{i=1}^n \alpha_i y_i \varphi(\boldsymbol{x}_i). \tag{3.32}$$

It is immediately to note that

$$\begin{aligned} \boldsymbol{w}^\top \varphi(\boldsymbol{x}) &= \left(-\eta \sum_{i=1}^n \alpha_i y_i \varphi(\boldsymbol{x}_i)\right)^\top \varphi(\boldsymbol{x}) \\ &= -\eta \sum_{i=1}^n \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}), \end{aligned}$$

and the offset b can be obtained as

$$\begin{aligned} b &= \boldsymbol{w}^\top \varphi(\boldsymbol{x}_s) - y_s \\ &= -\eta \sum_{i=1}^n \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}_s) - y_s \end{aligned}$$

where $\boldsymbol{x}_s$ is a support vector (i.e. such that $0 < \alpha_s < \frac{1}{2}\left((1-2\tau)\, y_s + 1\right)$).
Then, the kernel-based scalable SVM predictor classifier becomes

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = -\eta \sum_{i=1}^n \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) - b + \rho,$$

where the coefficient α_i are obtained by (3.32). □

3.5.2 Scalable SVDD

Proof. Again, we switch to the dual of problem (3.16). Defining the multipliers as in (3.25)-(3.26) the Lagrangian is given by

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}, R, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) =& \frac{1}{2\eta}R^2 + \frac{1}{2}\sum_{i=1}^n \left((1-2\tau)y_i + 1\right) \\ & - \sum_{i=1}^n \alpha_i \left[\xi_i + y_i\left(\|\varphi(\boldsymbol{x}_i) - \boldsymbol{w}\|^2 - R^2\right)\right] - \sum_{i=1}^n \beta_i \xi_i. \end{aligned} \tag{3.33}$$

Recalling that $\|\varphi(\boldsymbol{x}) - \boldsymbol{w}\|^2 = \|\varphi(\boldsymbol{x})\|^2 - 2\varphi(\boldsymbol{x})^\top \boldsymbol{w} + \|\boldsymbol{w}\|^2$ and setting partial derivatives to zero, we obtain the following constraints:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = 2\eta \sum_{i=1}^{n} \alpha_i y_i \varphi(\boldsymbol{x}_i) \tag{3.34}$$

$$\frac{\partial \mathcal{L}}{\partial R} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = -\frac{1}{2\eta}, \tag{3.35}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \Rightarrow \beta_i = -\alpha_i + \frac{1}{2}\left((1-2\tau)\, y_i + 1\right)$$

$$\Rightarrow 0 \leq \alpha_i \leq \frac{1}{2}\left((1-2\tau)\, y_i + 1\right)\ i \in [n]. \tag{3.36}$$

Substituting (3.34), (3.35) and (3.36) into (3.33) we obtain:

$$\mathcal{L} = \sum_{i=1}^{n} \alpha_i y_i \varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_i) - 2\eta \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_j),$$

and using the notation introduced in (7.13), (7.14) and (7.15) we can write $\mathcal{L}$ in a more compact form

$$\mathcal{L} = -2\eta \boldsymbol{\alpha}^\top D K D \boldsymbol{\alpha} + \boldsymbol{\alpha}^\top D \mathrm{diag}(K),$$

that has to be maximized with respect the dual variables α_i, $i \in [n]$ and (3.29),(3.30) constraints, i.e.

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & 2\eta \boldsymbol{\alpha}^\top D K D \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top D \mathrm{diag}(K) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = -\frac{1}{2\eta}, \\ & 0 \leq \alpha_i \leq \frac{1}{2}\left((1-2\tau)\, y_i + 1\right),\ i \in [n]. \end{aligned} \tag{3.37}$$

In particular, the center is obtained as

$$\boldsymbol{w} = 2\eta \sum_{i=1}^{n} \alpha_i y_i \varphi(\boldsymbol{x}_i). \tag{3.38}$$

It is valuable to note that

$$\begin{aligned} \|\varphi(\boldsymbol{x}) - \boldsymbol{w}\|^2 &= (\varphi(\boldsymbol{x}) - \boldsymbol{w})^\top (\varphi(\boldsymbol{x}) - \boldsymbol{w}) \\ &= \varphi(\boldsymbol{x})^\top \varphi(\boldsymbol{x}) - 2\varphi(\boldsymbol{x})^\top \boldsymbol{w} + \boldsymbol{w}^\top \boldsymbol{w} \\ &= K(\boldsymbol{x}, \boldsymbol{x}) - 4\eta \sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) \\ &\quad + 4\eta^2 \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j). \end{aligned}$$

The radius R is computed on the basis of the support vectors $\boldsymbol{x}_s$, feature points such that $0 < \alpha_s < \frac{1}{2}\left((1-2\tau)\, y_s + 1\right)$. Specifically,

$$R^2 = \left\|\varphi(\boldsymbol{x}_s) - \boldsymbol{w}\right\|^2.$$

Since the target points are those enclosed in the hypersphere (i.e. $\left\|\varphi(\boldsymbol{x}) - \boldsymbol{w}\right\|^2 \leq R^2$), we define the classifier predictor for the SVDD as

$$\begin{aligned} f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = K(\boldsymbol{x}, \boldsymbol{x}) - 4\eta \sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) \\ + 4\eta^2 \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) - (R^2 - \rho), \end{aligned}$$

where the coefficient α_i are obtained from (3.38). □

3.5.3 Scalable LR

Proof. Starting from (3.17), we introduce the variables

$$\begin{aligned} \xi_i &= -y_i \left(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - b\right) \\ C_i &= \frac{1}{2}\left((1-2\tau) y_i + 1\right) \end{aligned}$$

and define the following minimization problem

$$\begin{aligned} \min_{\boldsymbol{w}, b, \xi_1, \ldots, \xi_n} \quad & \frac{1}{2\eta} \boldsymbol{w}^\top \boldsymbol{w} + \sum_{i=1}^{n} C_i \log\left(1 + e^{\xi_i}\right) \\ \text{s.t.} \quad & \xi_i = -y_i (\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - b),\ i \in [n]. \end{aligned}$$

Also in this case, we can obtain the dual form by defining the multipliers $\boldsymbol{\alpha}$ as in (3.25). The Lagrangian is given by

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = & \frac{1}{2\eta} \boldsymbol{w}^\top \boldsymbol{w} + \sum_{i=1}^{n} C_i \log\left(1 + e^{-y_i\left(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - b\right)}\right) \\ & - \sum_{i=1}^{n} \alpha_i \left[\xi_i - y_i \left(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - b\right)\right]. \end{aligned}$$

Setting partial derivatives to zero, we obtain the following constraints:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = 2\eta \sum_{i=1}^{n} \alpha_i y_i \varphi(\boldsymbol{x}_i) \tag{3.39}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0, \tag{3.40}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \xi_i} = 0 &\Rightarrow C_i \frac{e^{\xi_i}}{1+e^{\xi_i}} - \alpha_i = 0 \\ &\Rightarrow (C_i - \alpha_i)\, e^{\xi_i} - \alpha_i = 0 \\ &\Rightarrow e^{\xi_i} = \frac{\alpha_i}{C_i - \alpha_i} \geq 0 \\ &\Rightarrow 0 \leq \alpha_i \leq C_i, \quad i \in [n]. \end{aligned} \tag{3.41}$$

Substituting (3.39), (3.40) and (3.41) we obtain:

$$\begin{aligned} \mathcal{L} = &- \frac{\eta}{2} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j \varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_j) - \sum_{i=1}^{n} \alpha_i \log(\alpha_i) \\ &- \sum_{i=1}^{n} (C_i - \alpha_i) \log(C_i - \alpha_i) + \sum_{i=1}^{n} C_i \log(C_i), \end{aligned}$$

and using the notation introduced in (7.13), (7.14) and (7.15) we can write $\mathcal{L}$ in a more compact form as

$$\begin{aligned} \mathcal{L} = &- \frac{\eta}{2} \boldsymbol{\alpha}^\top DKD\boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \log(\boldsymbol{\alpha}) \\ &- (\mathbf{C} - \boldsymbol{\alpha}) \log(\mathbf{C} - \boldsymbol{\alpha}) + \mathbf{C}^\top \log(\mathbf{C}), \end{aligned}$$

where $\mathbf{C} \doteq [C_1\, C_2\, \cdots\, C_n]^\top$. This function has to be maximized with respect the dual variables α_i, $i \in [n]$ and (3.40),(3.41) constraints. Since $\mathbf{C}^\top \log(\mathbf{C})$ is a constant and adding constants does not change the optimization problem, we can neglect it. So the dual formulation of the weighted logistic regression is

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{\eta}{2} \boldsymbol{\alpha}^\top DKD\boldsymbol{\alpha} + \boldsymbol{\alpha}^\top \log(\boldsymbol{\alpha}) - (\mathbf{C} - \boldsymbol{\alpha}) \log(\mathbf{C} - \boldsymbol{\alpha}) \\ \text{s.t.} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq \frac{1}{2}\left((1-2\tau)\, y_i + 1\right),\ i \in [n]. \end{aligned} \tag{3.42}$$

The vector of weights is obtained as

$$\boldsymbol{w} = \eta \sum_{i=1}^{n} \alpha_i y_i \varphi(\boldsymbol{x}_i), \tag{3.43}$$

and then (exactly as for SVM) we get

$$\boldsymbol{w}^\top \varphi(\boldsymbol{x}) = \eta \sum_{i=1}^{n} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}),$$

computing the offset b as for SVM by the support vectors $\boldsymbol{x}_s$. that are such that $0 < \alpha_s < \frac{1}{2}\left((1-2\tau)\, y_s + 1\right)$.
Finally, with a bit of algebra, the kernel formulation of the scalable predictor classifier for LR can be written as

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = \frac{1}{2} \frac{e^{-b-\rho} - \prod_{i=1}^{n} e^{-\eta \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x})}}{e^{-b-\rho} + \prod_{i=1}^{n} e^{-\eta \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x})}},$$

where the α_i are obtained from (3.43).

□

Chapter 4

Probabilistic Safety Regions for eXponential Distributions

In this work, I've tried to give an alternative definition of Probabilistic Safety Region, relying on the assumption of having data coming from an exponential distribution.

In this new contest, we define a *Probabilistic Safety Region* as

$$\Phi = \{\boldsymbol{x} : p(A|\boldsymbol{x}) \geq 1 - \varepsilon\}.$$

It outlines the set of samples $\boldsymbol{x}$ such that the probability of observing the event A given $\boldsymbol{x}$ is greater or equal than $1 - \varepsilon$, where ε denotes the confidence. In this paper we make a strong assumption that allows the PSR to be written as a compact set: we assume that the data belong to exponential distributions [172]. This assumption makes it easy to show that the PSR has a boundary $\Gamma(\boldsymbol{x}) + \gamma$, $\gamma \in \mathbb{R}$, uniquely determined by the training points and its size can be handled by a radius $\tilde{\rho} = \rho - \gamma \geq 0$ which depends from the confidence ε and the probability p_A of the event A:

$$\Phi = \{\boldsymbol{x} : \Gamma(\boldsymbol{x}) + \gamma \leq \rho(p_A, \varepsilon)\}.$$

With this assumption, it is possible to give operational meaning to the PSR, which, as will become clear later in the article, can be thought of as a true classifier. Specifically, we will show that the PSR can be modeled as a Support Vector Machine (SVM), thereby recovering all its good properties of robustness and implementability:

$$\tilde{\Phi} = \{\ \boldsymbol{x}\ :\ \boldsymbol{w}^\top \varphi(\boldsymbol{x}) - c \geq 0\ \},$$

where in a sense the offset c plays the role of the radius of the approximate PSR as detailed later on.

4.1 PSR for eXponential Distribution

We consider two complementary events A and B, with probabilities p_A and $p_B = 1 - p_A$. We consider a sample space Ω with probability distribution function $f(\boldsymbol{x})$. We assume that the events A and B have a probabilistic effect on $\boldsymbol{x}$. That is, we consider the density functions $f(\boldsymbol{x}|A)$ and $f(\boldsymbol{x}|B)$ that serve to characterize $f(\boldsymbol{x})$ in terms of A and B:

$$f(\boldsymbol{x}) = f(\boldsymbol{x}|A)p_A + f(\boldsymbol{x}|B)p_B.$$

We assume that $f(\boldsymbol{x}|A)$ and $f(\boldsymbol{x}|B)$ belong to an exponential family [172]. That is, for all $\boldsymbol{x} \in \Omega$,

$$\begin{aligned} f(\boldsymbol{x}|A) &= \frac{1}{c_A} \exp\left(-g_A(\boldsymbol{x})\right), \\ f(\boldsymbol{x}|B) &= \frac{1}{c_B} \exp\left(-g_B(\boldsymbol{x})\right), \end{aligned} \tag{4.1}$$

where g_A and g_B are measurable functions of $\boldsymbol{x}$. The normalising constants c_A and c_B are given by

$$\begin{aligned} c_A &= \int_{\Omega} \exp\left(-g_A(\boldsymbol{x})\right) d\boldsymbol{x}, \\ c_B &= \int_{\Omega} \exp\left(-g_B(\boldsymbol{x})\right) d\boldsymbol{x}. \end{aligned}$$

Denote $p(A|\boldsymbol{x})$ the probability of event A given $\boldsymbol{x}$. We now focus on the notion of *safety region*, i.e. the region $\Phi_{p_A,\varepsilon}$ of Ω for which the probability of event A, given $\boldsymbol{x}$ is larger than $1 - \varepsilon$, where $\varepsilon \in (0, 1)$ is a probabilistic parameter. That is,

Definition 4.1 (Probabilistic Safety Region for eXponential Distribution)**.** Given $p_A \in (0, 1)$, $\varepsilon \in (0, 1)$ and the density function $f(\boldsymbol{x}|A)$, we define the *Probabilistic Safety Region* (PSR) relative to A as

$$\Phi_{p_A,\varepsilon} = \{\boldsymbol{x} : p(A|\boldsymbol{x}) \geq 1 - \varepsilon\}.$$

Since we are considering exponential density functions, it is possible to write $\Phi_{p_A,\varepsilon}$ in a more operational form, as it is shown in the following proposition.

Proposition 4.2 (Scaling form of PSR)**.** *Suppose that $f(\boldsymbol{x}|A)$ and $f(\boldsymbol{x}|B)$ are of the form* (4.1)*. Then, given $p_A \in (0, 1)$ and $\varepsilon \in (0, 1)$ the PSR can be written as*

$$\Phi_{p_A,\varepsilon} = \Big\{ \boldsymbol{x} : \Gamma(\boldsymbol{x}) + \gamma \leq \rho(p_A, \varepsilon) \Big\}$$

where

$$\begin{aligned}\Gamma(\boldsymbol{x}) &= g_A(\boldsymbol{x}) - g_B(\boldsymbol{x}),\\ \gamma &= \ln\frac{c_A}{c_B},\\ \rho(p_A,\varepsilon) &= \rho_{p_A} + \rho_\varepsilon,\\ \rho_{p_A} &= \ln\frac{p_A}{1-p_A},\\ \rho_\varepsilon &= \ln\frac{\varepsilon}{1-\varepsilon}.\end{aligned}$$

Proof. It is clear that, by definition, $\boldsymbol{x} \in \Phi_{p_A,\varepsilon}$ implies that event A occurs for $\boldsymbol{x}$ with probability at least $1-\varepsilon$. Thus,

$$p(B|\boldsymbol{x}) = 1 - p(A|\boldsymbol{x}) \leq \varepsilon,\ \forall \boldsymbol{x} \in \Phi_{p_A,\varepsilon}.$$

Given $p_A \in (0,1)$, $\varepsilon \in (0,1)$ and the density function $f(\boldsymbol{x}|A)$, we can compute the PSR from

$$p(A|\boldsymbol{x})f(\boldsymbol{x}) = f(\boldsymbol{x}|A)p_A. \tag{4.2}$$

Thus,

$$p(A|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|A)p_A}{f(\boldsymbol{x})} = \frac{f(\boldsymbol{x}|A)p_A}{f(\boldsymbol{x}|A)p_A + f(\boldsymbol{x}|B)p_B}.$$

Since $\dfrac{z}{1-z}$ is monotonically growing in $[0,1)$,

$$p(A|\boldsymbol{x}) \geq 1-\varepsilon \iff \frac{p(A|\boldsymbol{x})}{1-p(A|\boldsymbol{x})} \geq \frac{1-\varepsilon}{\varepsilon}. \tag{4.3}$$

From (4.2) we have

$$p(A|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|A)p_A}{f(\boldsymbol{x})}.$$

Similarly,

$$p(B|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|B)p_B}{f(\boldsymbol{x})}.$$

Thus, we get

$$\frac{p(A|\boldsymbol{x})}{1-p(A|\boldsymbol{x})} = \frac{p(A|\boldsymbol{x})}{p(B|\boldsymbol{x})} = \frac{f(\boldsymbol{x}|A)p_A}{f(\boldsymbol{x}|B)p_B}.$$

This, along with (4.3), yields

$$p(A|\boldsymbol{x}) \geq 1-\varepsilon \iff \frac{f(\boldsymbol{x}|A)p_A}{f(\boldsymbol{x}|B)p_B} \geq \frac{1-\varepsilon}{\varepsilon}$$

Then,

$$\begin{aligned}
\Phi_{p_A,\varepsilon} &= \left\{\boldsymbol{x} : p(A|\boldsymbol{x}) \geq 1-\varepsilon\right\} \\
&= \left\{\boldsymbol{x} : f(\boldsymbol{x}|A) \geq \left(\frac{1-\varepsilon}{\varepsilon}\right)\frac{p_B}{p_A} f(\boldsymbol{x}|B)\right\} \\
&= \left\{\boldsymbol{x} : f(\boldsymbol{x}|A) \geq \left(\frac{1-\varepsilon}{\varepsilon}\right)\left(\frac{1-p_A}{p_A}\right) f(\boldsymbol{x}|B)\right\} \\
&= \left\{\boldsymbol{x} : f(\boldsymbol{x}|A) \geq \exp(-\rho(p_A,\varepsilon)) f(\boldsymbol{x}|B)\right\},
\end{aligned}$$

and it is easy to see that

$$\begin{aligned}
\rho(p_A,\varepsilon) &= \ln\left(\frac{\varepsilon}{1-\varepsilon}\frac{p_A}{1-p_A}\right) \\
&= \ln\frac{\varepsilon}{1-\varepsilon} + \ln\frac{p_A}{1-p_A} \\
&= \rho_{p_A} + \rho_\varepsilon
\end{aligned}$$

Taking logarithms and using (4.2)

$$\begin{aligned}
\Phi_{p_A,\varepsilon} &= \Big\{\boldsymbol{x} : -g_A(\boldsymbol{x}) - \ln\, c_A \geq \\
&\qquad -\rho(p_A,\varepsilon) - g_B(\boldsymbol{x}) - \ln\, c_B\Big\} \\
&= \left\{\boldsymbol{x} : g_A(\boldsymbol{x}) - g_B(\boldsymbol{x}) + \ln\,\frac{c_A}{c_B} \leq \rho(p_A,\varepsilon)\right\}.
\end{aligned}$$

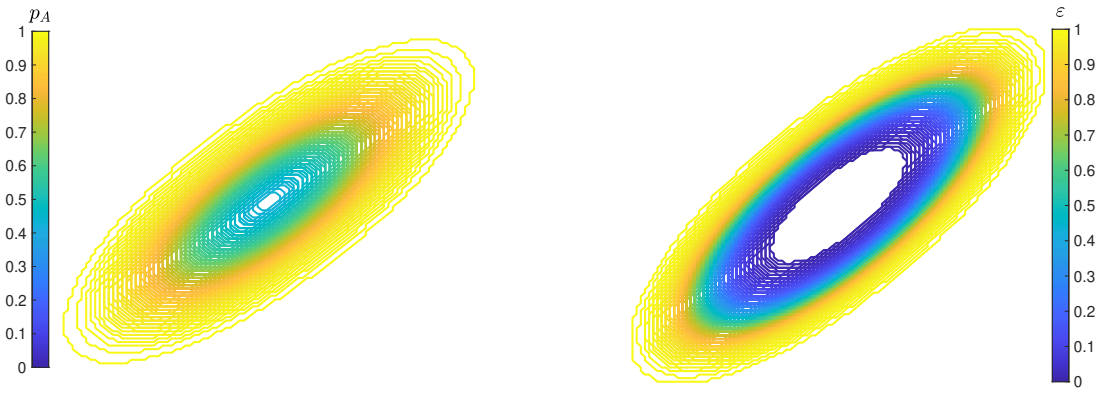
□

Proposition 4.2 means that the PSR for exponential distributions is controllable by a radius depending on probabilistic parameters p_A and ε. That is, once its shape has been uniquely determined by the observations $\boldsymbol{x}$ through $\Gamma(\boldsymbol{x})$ (eventually shifted by a factor γ), it is sufficient to vary the radius $\rho(p_A,\varepsilon)$ for getting the desired amount of confidence ε of being in $\Phi_{p_A,\varepsilon}$ with probability p_A.

Remark 4.3 (PSR form for normal distribution)**.** Focusing on a specific exponential family, the normal distribution, we can give an exact description of the PSR. Suppose now that $f(\boldsymbol{x}|A)$ and $f(\boldsymbol{x}|B)$ are Gaussian, that is:

$$\begin{aligned}
f(\boldsymbol{x}|A) &= \frac{1}{c_A}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_A)^\top \Sigma_A^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_A)\right), \\
f(\boldsymbol{x}|B) &= \frac{1}{c_B}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_B)^\top \Sigma_B^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_B)\right)
\end{aligned}$$

where, respectively for A and B



(a) PSR for Gaussian distribution varying p_A and leaving ε fixed at 0.1.

(b) PSR for Gaussian distribution varying ε and leaving p_A fixed at 0.8.

Figure 4.1: This figure shows the shape of the PSR as the radius changes, as a function of ε and p_A. The key point to note is that the shape of the region remains the same and is only scaled (enlarged or reduced) according to the radius.

$$c_A = \sqrt{\det(2\pi\Sigma_A)} \text{ and } c_B = \sqrt{\det(2\pi\Sigma_B)},$$

$\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ are the means and Σ_A and Σ_B are the covariance matrices.
This corresponds to

$$\begin{aligned} g_A(\boldsymbol{x}) &= \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_A)^\top \Sigma_A^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_A), \\ g_B(\boldsymbol{x}) &= \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_B)^\top \Sigma_B^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_B), \\ \gamma &= \ln \frac{\det(\Sigma_A)}{\det(\Sigma_B)}. \end{aligned}$$

Taking into account Proposition 4.2 and exploiting that the sum of quadratic forms is again a quadratic since the covariance matrix is symmetric, we obtain that the PSR for the normal distribution is an ellipse:

$$\Phi_{p_A,\varepsilon} = \left\{ \boldsymbol{x} : \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma(\boldsymbol{x}-\boldsymbol{\mu}) + \gamma' \leq \rho\left(p_A, \varepsilon\right) \right\},$$

where

$$\begin{aligned} \Sigma &= \Sigma_A^{-1} - \Sigma_B^{-1}, \\ \boldsymbol{\mu} &= \Sigma^{-1}\left(\Sigma_A^{-1}\boldsymbol{\mu}_A - \Sigma_B^{-1}\boldsymbol{\mu}_B\right), \\ \gamma' &= -\gamma - \boldsymbol{\mu}_A^\top \Sigma_A^{-1}\boldsymbol{\mu}_A + \boldsymbol{\mu}_B^\top \Sigma_B^{-1}\boldsymbol{\mu}_B + \boldsymbol{\mu}^\top \Sigma \boldsymbol{\mu}. \end{aligned}$$

This example (and in particular the Figure 4.1[1] is quite explanatory) gives us the opportunity to focus on the fact that the PSR for the exponential distribution is controllable by the radius $\rho(p_A, \varepsilon)$ and that its shape does not depend on the values of the probability p_A and confidence ε but only on the data $\boldsymbol{x}$. Thus, fixed the data, the shape is fixed.
More can be said about the role of p_A and ε. Thanks to the beautiful property that $\rho(p_A, \varepsilon) = \rho_{p_A} + \rho_\varepsilon$ the contribution of the parameters is independent of each other, so fixed probability the PSR can be controlled only by confidence or vice versa. Again from the Figure 4.1, it is easy to confirm that smaller values of the parameters p_A, ε mean smaller PSRs: if one wants to be in $\Phi_{p_A,\varepsilon}$ with low probability or high confidence, the price to be paid is that the PSR is not too large. In a sense, we are trying to minimize false positives, that is, those observations that belong to event B but are actually included in A. Of course, smaller the region, lower the probability of making errors. This comment allows us to introduce readers to the fact that the PSR can be learned when the distribution is not known. In other words, we want to say that the problem of defining a PSR can be addressed by exploiting information from the data. In particular, as we will see later in the discussion, once the shape of the PSR is learned, it is sufficient to vary the radius to scale the region to the desired confidence or probability values.

4.2 SVM based approximations of the safety region

Suppose that we have a collection of labelled data points

$$(\boldsymbol{x}_i, y_i),\ i \in [n],$$

where

$$y_i = \begin{cases} +1 \text{ if } \boldsymbol{x}_i \text{ has label } A \\ -1 \text{ if } \boldsymbol{x}_i \text{ has label } B. \end{cases}$$

Our goal is to find a good approximation of the safety region $\Phi_{p_A,\varepsilon}$. To this end, we introduce a parameterization of $g_A(\boldsymbol{x})$ and $g_B(\boldsymbol{x})$ as follows:

$$\begin{aligned} -g_A(\boldsymbol{x}) &\approx \boldsymbol{w}_A^\top \varphi(\boldsymbol{x}), \\ -g_B(\boldsymbol{x}) &\approx \boldsymbol{w}_B^\top \varphi(\boldsymbol{x}). \end{aligned}$$

We notice that the expression for the safety region depends on the difference of $g_A(\boldsymbol{x})$ and $g_B(\boldsymbol{x})$, that is, on $(\boldsymbol{w}_A - \boldsymbol{w}_B)^\top \varphi(\boldsymbol{x})$. Thus, in order to obtain/estimate the safety region it suffices to consider the difference $\boldsymbol{w} = \boldsymbol{w}_A - \boldsymbol{w}_B$:

$$\Phi_{p_A,\varepsilon} \approx \{\ \boldsymbol{x}\ :\ -\boldsymbol{w}^\top \varphi(\boldsymbol{x}) + \gamma \leq \rho(p_A, \varepsilon)\ \}$$

[1]For the sake of completeness, the means and covariance matrices of the Gaussians in Figure 1 are for A and B, respectively $\boldsymbol{\mu}_A = [0,0]^\top, \Sigma_A = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$ and $\boldsymbol{\mu}_B = [1,1]^\top, \Sigma_B = \begin{bmatrix} 2 & 0.1 \\ 0.1 & 2 \end{bmatrix}$.

Then defining

$$c = \gamma - \rho(p_A, \varepsilon)$$

we have

$$\tilde{\Phi} = \{\ \boldsymbol{x}\ :\ \boldsymbol{w}^\top \varphi(\boldsymbol{x}) - c \geq 0\ \} \tag{4.4}$$

The main contribution of equation (4.4) is to show that if

$$\begin{aligned} f(\boldsymbol{x}|A) &= \frac{1}{c_A} \exp\left(\boldsymbol{w}_A^\top \varphi(\boldsymbol{x})\right) \\ f(\boldsymbol{x}|B) &= \frac{1}{c_B} \exp\left(\boldsymbol{w}_B^\top \varphi(\boldsymbol{x})\right), \end{aligned}$$

then the probabilistic safety region is given by level sets of $\boldsymbol{w}^\top \varphi(x)$, where $\boldsymbol{w}$ does not depend on p_A or ε.

We now present a first (naive) approach for the computation of $\mathbf{w}$ and c.

We would like $\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - c$ to be positive if $\boldsymbol{x}_i$ is labelled as A (i.e. $y_i = +1$), and negative otherwise. That is, we would like the quantity $y_i(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - c)$ to be positive with high probability in every situation. Based on this, we now formulate an optimisation problem leading to a classifier distinguishing between classes A and B.

$$\begin{aligned} \min_{\boldsymbol{w}, c, \xi_1, \ldots, \xi_n} \quad & \frac{1}{2\eta} \boldsymbol{w}^\top \boldsymbol{w} + \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - c) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned}$$

The hyper-parameter $\eta > 0$ serves to make a trade-off between regularisation and missclassification error. Once the value of $\boldsymbol{w}$ and c has been obtained, we could provide the following classifier:

$$\hat{y}(x) = \begin{cases} +1 & \text{if } \boldsymbol{w}^\top \varphi(\boldsymbol{x}) - c \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

Let us now denote as *false positive* the situation in which $\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - c$ is positive when the label of $\boldsymbol{x}_i$ is B and *false negative* when $\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - c$ is negative when $\boldsymbol{x}_i$ is labelled A.

The previous formulation penalises in the same way both missclassification errors. In order to cope with this, we introduce a weighting parameter $\tau \in (0, 1)$, and we formulate the weighted problem

$$\begin{aligned} \min_{\boldsymbol{w}, c, \xi_1, \ldots, \xi_n} \quad & \frac{1}{2\eta} \boldsymbol{w}^\top \boldsymbol{w} + \frac{1}{2} \sum_{i=1}^{n} \left((1 - 2\tau) y_i + 1\right) \xi_i \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - c) \geq 1 - \xi_i,\ i \in [n], \\ & \xi_i \geq 0,\ i \in [n]. \end{aligned}$$

It is clear that values of τ close to zero will penalise quite a lot the false negative errors and very little the positive errors. On the other hand, values of τ close to one yield just the opposite behaviour. The value of τ is then related to the false positive rate. As a matter of fact, it is well known in the quantile regression literature [21], that if one discards the regularisation term (i.e. $\eta \to \infty$) then the false negative ratio tends to τ when the number of samples tends to infinity (under not very restrictive assumptions on the data).
However, when η is not a large value, it is not that simple to relate τ with the false negative ratio. Only a qualitative relationship exists.
From the previous discussion we infer that the optimal parameter $\boldsymbol{w}$ does not depend on the particular specifications for ε, or the value of p_A. Thus, a reasonable scheme would be to obtain the value of $\boldsymbol{w}$ that better fits different values of τ. That is, one formulates an optimisation problem in which we consider different values τ_k with its corresponding optimal values c_k, but all of them sharing the same value for $\boldsymbol{w}$. That is, given $\eta > 0$ we introduce the set of positive values $\{\tau_1, \tau_2, \ldots, \tau_m\}$. Then, for each $i = 1, \ldots, n$ and $k = 1, \ldots, m$, we let

$$\xi_{i,k} \doteq \max\big(0, y_i(\boldsymbol{w}^T \varphi(\boldsymbol{x}_i) - c_k)\big).$$

and thus the decision variables

$$\boldsymbol{w},\ \mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} \quad \boldsymbol{\xi}_i = \begin{bmatrix} \xi_{i,1} \\ \xi_{i,2} \\ \vdots \\ \xi_{i,m} \end{bmatrix},\ i \in [n].$$

The resulting optimization problem is

$$\begin{aligned} \min_{\boldsymbol{w},\mathbf{c},\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_n} \ & \frac{1}{2\eta}\boldsymbol{w}^\top \boldsymbol{w} + \\ & \frac{1}{2}\sum_{k=1}^{m}\sum_{i=1}^{n} \left((1-2\tau_k)y_i + 1\right)\xi_{i,k} \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - c_k) \geq 1 - \xi_{i,k}, \\ & \xi_{i,k} \geq 0, \\ & i \in [n], k \in [m]. \end{aligned} \tag{4.5}$$

Then, we want to solve a system composed by m SVMs with the same data $\{\boldsymbol{x}_i\}_{i=1}^n$, with a unique hyperplane $\boldsymbol{w}$ but different offsets c_k and different weights τ_k, $k = 1, \ldots, m$.
In order to derive the Lagrangian dual of (4.5) we recall

$$\begin{aligned} R &= \left[\ \varphi(\boldsymbol{x}_1) \quad \varphi(\boldsymbol{x}_2) \quad \ldots \quad \varphi(\boldsymbol{x}_n)\ \right], \\ D &= \operatorname{diag}\{y_1, y_2, \ldots, y_n\}, \\ K &= RR^\top, \end{aligned}$$

where $K_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_j),\ i \in [n], j \in [n]$, is the kernel matrix.

Proposition 4.4 (Dual form of SSVM)**.** *Denoted with $\alpha_{i,k}$ the $i-th$ Lagrange multiplier relative to the $k-th$ SVM, with $\bar{\alpha}_i = \sum_{k=1}^{m} \alpha_{i,k}$ and defined*

$$\bar{\boldsymbol{\alpha}} = \begin{bmatrix} \bar{\alpha}_1 \\ \bar{\alpha}_2 \\ \vdots \\ \bar{\alpha}_n \end{bmatrix},$$

the dual form of (4.5) *is*

$$\begin{aligned} \max_{\bar{\boldsymbol{\alpha}}, \bar{\alpha_1}, \ldots, \bar{\alpha_n}} \quad & -\frac{\eta}{2} \bar{\boldsymbol{\alpha}}^\top D K D \bar{\boldsymbol{\alpha}} + \sum_{i=1}^{n} \bar{\alpha}_i \\ s.t. \quad & \bar{\alpha}_i = \sum_{k=1}^{n} \alpha_{i,k}, \ i \in [n], \\ & \sum_{i=1}^{n} \alpha_{i,k} y_i = 0, \ k \in [m], \\ & 0 \leq \alpha_{i,k} \leq \frac{1}{2} \left((1 - 2\tau_k) y_i + 1 \right), \\ & i \in [n], \ k \in [m]. \end{aligned}$$

Proof. Defining the multipliers

$$\boldsymbol{\alpha}_i = \begin{bmatrix} \alpha_{i,1} \\ \alpha_{i,2} \\ \vdots \\ \alpha_{i,m} \end{bmatrix} \quad \boldsymbol{\beta}_i = \begin{bmatrix} \beta_{i,1} \\ \beta_{i,2} \\ \vdots \\ \beta_{i,m} \end{bmatrix}, \ i \in [n],$$

the Lagrangian is given by

$$\begin{aligned} \mathcal{L} =& \mathcal{L}(\boldsymbol{w}, \mathbf{c}, \boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_n) \\ =& \frac{1}{2\eta} \boldsymbol{w}^\top \boldsymbol{w} + \frac{1}{2} \sum_{k=1}^{m} \sum_{i=1}^{n} \left((1 - 2\tau_k) y_i + 1 \right) \xi_{i,k} \\ & - \sum_{i=1}^{n} \sum_{k=1}^{m} \left(\alpha_{i,k} \left(y_i (\boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) - c_k) - 1 + \xi_{i,k} \right) + \beta_{i,k} \xi_{i,k} \right) \\ =& \frac{1}{2\eta} \boldsymbol{w}^\top \boldsymbol{w} + \frac{1}{2} \sum_{k=1}^{m} \sum_{i=1}^{n} \left((1 - 2\tau_k) y_i + 1 \right) \xi_{i,k} \\ & - \sum_{i=1}^{n} \left(\sum_{k=1}^{m} \alpha_{i,k} \right) y_i \boldsymbol{w}^\top \varphi(\boldsymbol{x}_i) \\ & - \sum_{i=1}^{n} \sum_{k=1}^{m} \left(\alpha_{i,k} \left(-y_i c_k - 1 + \xi_{i,k} \right) + \beta_{i,k} \xi_{i,k} \right) \end{aligned}$$

Let us define

$$\bar{\alpha}_i = \sum_{k=1}^{m} \alpha_{i,k}.$$

We obtain

$$\begin{aligned}
\mathcal{L} &= \frac{1}{2\eta}\boldsymbol{w}^\top \boldsymbol{w} + \\
&\frac{1}{2}\sum_{k=1}^{m}\sum_{i=1}^{n}\left((1-2\tau_k)y_i + 1 - 2\alpha_{i,k} - 2\beta_{i,k})\right)\xi_{i,k} - \\
&\left(\sum_{i=1}^{n}\bar{\alpha}_i y_i \varphi(x_i)\right)^\top \boldsymbol{w} - \sum_{i=1}^{n}\bar{\alpha}_i + \sum_{i=1}^{n}\sum_{k=1}^{m}\alpha_{i,k} y_i c_k.
\end{aligned}$$

With this notation, and setting partial derivatives to zero gives the following constraints:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = 0 \Rightarrow \boldsymbol{w} = \eta \sum_{i=1}^{n}\bar{\alpha}_i y_i \varphi(\boldsymbol{x}_i) \tag{4.6}$$

$$\frac{\partial \mathcal{L}}{\partial c_k} = 0 \Rightarrow \sum_{i=1}^{n}\alpha_{i,k} y_i = 0,\ k \in [m] \tag{4.7}$$

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \xi_{i,k}} = 0 \Rightarrow \beta_{i,k} &= -\alpha_{i,k} + \frac{1}{2}\left((1-2\tau_k)\, y_i + 1\right) \\
&\Rightarrow 0 \leq \alpha_{i,k} \leq \frac{1}{2}\left((1-2\tau_k)\, y_i + 1\right) \qquad (4.8) \\
&\quad i \in [n],\ k \in [m].
\end{aligned}$$

Substituting (4.6) into (4.2) we obtain:

$$\begin{aligned}
\mathcal{L} &= \mathcal{L}(\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n, \bar{\alpha}_1, \ldots, \bar{\alpha}_n) \\
&= -\frac{\eta}{2}\left(\sum_{i=1}^{n} y_i \bar{\alpha}_i \varphi(\boldsymbol{x}_i)\right)^\top \left(\sum_{i=1}^{n} y_i \bar{\alpha}_i \varphi(\boldsymbol{x}_i)\right) + \sum_{i=1}^{n}\bar{\alpha}_i.
\end{aligned}$$

The objective is to maximise with respect the dual variables $\alpha_{i,k}$, $i \in [n]$, $k \in [m]$, $\bar{\alpha}_i \in [n]$ the Lagrangian $\mathcal{L}$ subject to the constraints

$$\begin{aligned}
\bar{\alpha}_i &= \sum_{k=1}^{n}\alpha_{i,k},\ i \in [n], \\
0 &= \sum_{i=1}^{n}\alpha_{i,k} y_i,\ k \in [m], \\
0 &\leq \alpha_{i,k} \leq \frac{1}{2}\left((1-2\tau_k)\, y_i + 1\right),\ i \in [n],\ k \in [m].
\end{aligned}$$

Let us recall

$$R = \left[\begin{array}{cccc} \varphi(\boldsymbol{x}_1) & \varphi(\boldsymbol{x}_2) & \dots & \varphi(\boldsymbol{x}_n) \end{array} \right], \tag{4.9}$$

$$\bar{\boldsymbol{\alpha}} = \left[\begin{array}{c} \bar{\alpha}_1 \\ \bar{\alpha}_2 \\ \vdots \\ \bar{\alpha}_n \end{array} \right], \tag{4.10}$$

$$D = \operatorname{diag}\{y_1, y_2, \dots, y_n\}. \tag{4.11}$$

With this notation,

$$\mathcal{L} = -\frac{\eta}{2}\bar{\boldsymbol{\alpha}}^\top DRR^\top D\bar{\boldsymbol{\alpha}} + \sum_{i=1}^{n} \bar{\alpha}_i.$$

Since

$$[RR^\top]_{i,j} = \varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K_{i,j},\ i \in [n], j \in [n],$$

we obtain

$$\mathcal{L} = -\frac{\eta}{2}\bar{\boldsymbol{\alpha}}^\top DKD\bar{\boldsymbol{\alpha}}^\top + \sum_{i=1}^{n} \bar{\alpha}_i,$$

where K is the kernel matrix, i.e. $K_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $i \in [n]$, $j \in [n]$. □

We notice that

$$\begin{aligned} \boldsymbol{w}^\top \varphi(\boldsymbol{x}) &= \left(\eta \sum_{i=1}^{n} \bar{\alpha}_i y_i \varphi(\boldsymbol{x}_i) \right)^\top \varphi(\boldsymbol{x}) \\ &= \eta \sum_{i=1}^{n} \bar{\alpha}_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}). \end{aligned}$$

Thus, once c has been chosen, a test point $\boldsymbol{x}$ would be classified as A if

$$\eta \sum_{i=1}^{n} \bar{\alpha}_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) - c \geq 0.$$

As far as the computational cost is concerned, since the SSVM is closely related to the SVM, it can be estimated similarly to that of the SVM [28] which, denoting by n the number of points and by d the number of features, is estimated as $O(\max(n, d) \cdot \min(n, d)^2)$. Of course, we have to take into account the complexity given by the weights stored in $\boldsymbol{\tau}$: if we denote by t the number of weights that are supposed to be used in computing the SSVM, the total cost of the SSVM will be $O(t \cdot (\max(n, d) \cdot \min(n, d)^2))$, since they appear as a summation in the minimization of the expected loss.

4.2.1 Probabilistic scaling for the choice of c

Once we have trained the SSVM and defined $\mathbf{w}^\top\varphi(\boldsymbol{x})$, what is missing is to choose appropriately the parameter c. Let us keep in mind that our goal is to provide a probabilistic bound on the prediction, i.e., we want to be sure that in the test phase the probability for the target class of being in the SVM approximation of the PSR $\tilde{\Phi}$ is greater than $1-\varepsilon$, with ε small. Thus, since we can think of c as the "radius" of our PSR (indeed it is, see (4.4)), we simply need to *scale* c to the value that best fits the desired confidence for the PSR.
We found that the best way in the literature to do this is again Probabilistic Scaling (see Chapter 3) and in order to apply it to this new case, first we have to introduce the following Proposition, which proves that a descending sequence of values for c determines a decreasing sequence of sets for the PSR.

Proposition 4.5. *Let $(c_i)_{i=1}^N$ a descending succession of real numbers, $c_1 > c_2 > \cdots > c_N$, and $\tilde{\Phi}_i = \{\boldsymbol{x} : \mathbf{w}^\top\varphi(\boldsymbol{x}) - c_i > 0\}$ for all $i = 1, \ldots, N$. Then also $\left(\tilde{\Phi}_i\right)_{i=1}^N$ is a descending succession of boxed sets:*

$$\tilde{\Phi}_1 \supset \tilde{\Phi}_2 \supset \cdots \supset \tilde{\Phi}_N.$$

Proof. Let $\boldsymbol{x} \in \tilde{\Phi}_1$. This means

$$\mathbf{w}^\top\varphi(\boldsymbol{x}) - c_1 > 0.$$

Since $c_1 > c_2 > \cdots > c_N$ then it holds that $-c_1 < -c_2 < \cdots < -c_N$ and then

$$0 < \mathbf{w}^\top\varphi(\boldsymbol{x}) - c_1 < \mathbf{w}^\top\varphi(\boldsymbol{x}) - c_2 < \cdots < \mathbf{w}^\top\varphi(\boldsymbol{x}) - c_N.$$

Then, the sets are boxed in a descending sequence. □

What we want to do is to choose c such that the SVM approximation $\tilde{\Phi}$ of the PSR $\Phi_{p_A,\varepsilon}$ has the desired level of confidence, i.e.

$$\mathbb{P}_\Omega\left(\boldsymbol{x} \text{ has label } A | \boldsymbol{x} \in \tilde{\Phi}\right) \geq 1-\varepsilon.$$

From the well established result [75, Property 3], already cited in the previous chapter of this thesis, we can provide such probabilistic bound for the approximation $\tilde{\Phi}$ of the PSR. Specifically we indicate with $\tilde{\Phi}_r^+$ the r-largest set among $\left(\tilde{\Phi}_i\right)_{i=1}^N$, that is there are no more than $r-1$ sets which contain $\tilde{\Phi}_r$. Hence, $\tilde{\Phi}_r^+$ is simply the r-th set of the sequence, since they are all boxed together.
Given $\varepsilon \in (0,1)$, $\delta \in (0,1)$,

$$N \geq \frac{7.47}{\varepsilon}\ln\frac{1}{\delta},\ r = \left\lfloor \frac{\varepsilon N}{2} \right\rfloor \tag{4.12}$$

and drawn N i.i.d. samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, then with a probability no smaller than $1-\delta$

$$\Pr_\Omega \left(\boldsymbol{x} \text{ has label } A \,|\, \boldsymbol{x} \in \tilde{\Phi}_r^+ \right) \geq 1 - \varepsilon.$$

So, now let's focus on how to choose c appropriately for the SSVM. Suppose we have N points of which N_{tr} are for training and N_{vl}, obtained from (4.12), are for choosing c. So $N_{tr} = N - N_{vl}$. After that the SSVM has been trained (i.e. $\Gamma(\boldsymbol{x}) = \mathbf{w}^\top \varphi(\boldsymbol{x})$ has been defined), for each point $\boldsymbol{x}_j$ of the validation set $X_{vl} \times Y_{vl} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{N_{vl}}$ we compute the offset c_j such that the decision boundary passes exactly through c_j, that is

$$c_j = \mathbf{w}^\top \varphi(\boldsymbol{x}_j)$$

for all $j = 1, \ldots, N_{vl}$. The c_j are then descending ordered and the r-th largest value, indicated in the following as c^*, is chosen according to (4.12). Then

$$\Phi^* = \{\boldsymbol{x} : \mathbf{w}^\top \varphi(\boldsymbol{x}) - c^* > 0\}$$

is the SVM approximation of Φ which satisfies the probabilistic requirements, that is

$$\Pr_\Omega \left(\boldsymbol{x} \text{ has label } A \,|\, \boldsymbol{x} \in \Phi^* \right) \geq 1 - \varepsilon.$$

4.3 Experiments

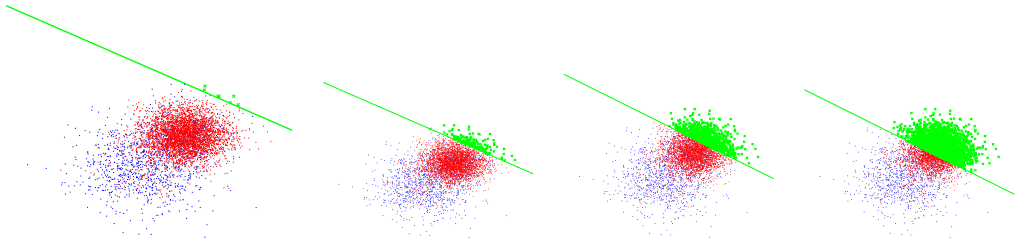
In this section, to test the efficiency of the SSVM and its ability to approximate the exact PSR, we report the numerical experiments in two relevant settings: simple 2D Gaussian case to better understand how the method works and multidimensional case, in which distributions other than Gaussian are tested. Then comparisons with the classic SVM are presented.

4.3.1 Easy case

Let us focus on a 2-dimensional toy problem to clearly understand the theory presented so far. We want to separate two classes of points, A and B, sampled by Gaussian distributions with respectively means and covariance matrices

$$\boldsymbol{\mu}_A = \begin{bmatrix} +1 \\ +1 \end{bmatrix}, \Sigma_A = I \; ; \; \boldsymbol{\mu}_B = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \Sigma_B = \frac{1}{2} I$$

where I is the identity matrix.
We sampled points with a probability $p_A = 0.8$ to belong to class A, so that the classification is unbalanced, and with probability $p_O = 0.1$ to get an outlier for each class, to add noise to the data. With these parameters, we constructed three datasets:

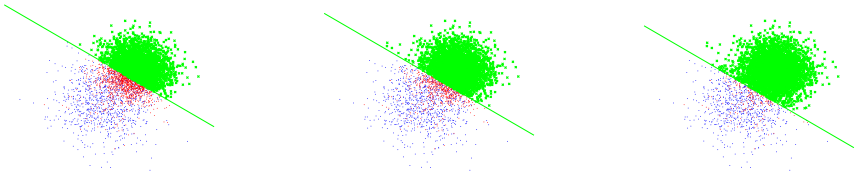


(a) $\varepsilon = 0.01$
$\mathbb{P}_\Omega(A|\Phi^*) = 1$

(b) $\varepsilon = 0.05$
$\mathbb{P}_\Omega(A|\Phi^*) = 0.97$

(c) $\varepsilon = 0.1$
$\mathbb{P}_\Omega(A|\Phi^*) = 0.91$

(d) $\varepsilon = 0.2$
$\mathbb{P}_\Omega(A|\Phi^*) = 0.88$

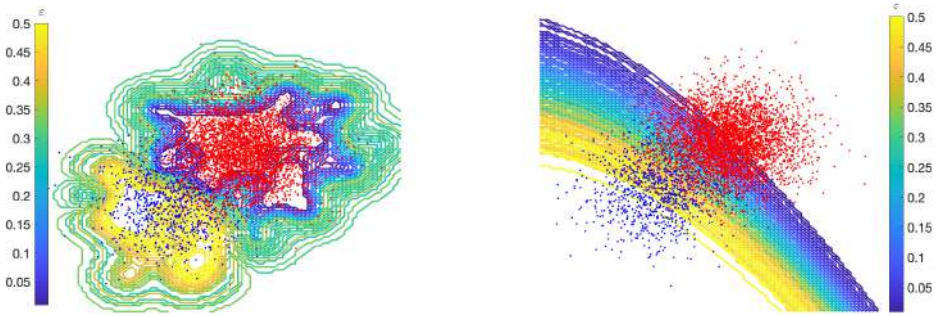


(e) $\varepsilon = 0.3$
$\mathbb{P}_\Omega(A|\Phi^*) = 0.89$

(f) $\varepsilon = 0.4$
$\mathbb{P}_\Omega(A|\Phi^*) = 0.87$

(g) $\varepsilon = 0.5$
$\mathbb{P}_\Omega(A|\Phi^*) = 0.85$

Figure 4.2: SSVM with linear kernel and different values of the confidence $1 - \varepsilon$. The points belonging to the safety region have been plotted in green.



(a) SSVM with Gaussian kernel.

(b) SSVM with quadratic kernel.

Figure 4.3: Behavior of the SSVM with nonlinear kernels. Colored bars show the dependence of PSR on ε, blue shades correspond to small values of ε, yellow shades to larger values.

- the training set X_{tr} with which we obtain the weights $\bar{\alpha}_i$ and thus $\Gamma(\boldsymbol{x}) = \mathbf{w}^\top \varphi(\boldsymbol{x})$. Its dimension is $N_{tr} = 2000$ points;
- the validation set X_{vl} with which to scale the value of c, according to confidence $1 - \varepsilon$. Its dimension N_{vl} is computed according to (4.12);
- the test set X_{ts} on which we evaluated the SSVM. Its dimension is $N_{ts} = 10000$ points.

The parameter of regularization η has been set at 1 and the weighting parameter

$$\boldsymbol{\tau} = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9].$$

The parameter δ for the scaling has been fixed to 10^{-6}.
For simplicity, we use a more compact notation for $\Pr_\Omega(\boldsymbol{x}$ has label $A \,|\, \boldsymbol{x} \in \Phi^*)$, namely

$$\Pr_\Omega(A|\Phi^*).$$

Figures 4.2 and 4.3 show the behaviour of the SSVM as a function of ε and the kernel. The red points belong to class A and the blue points belong to class B.
Specifically, Figure 4.2 describes the evolution of Φ^* for increasing values of ε with the linear kernel (for better visualization, the green crosses are for the test set points belonging to the PSR): (4.2a) represents the most conservative region since we want the probability of observing A from a point belonging to the PSR to be at least $1-0.01 = 0.99$ and on the other hand the PSR of (4.2g) allows for more uncertainty since the confidence should be greater than 0.5. Below each graph is the probability $\Pr_\Omega(A|\Phi^*)$ and, as it should be, its value is always greater than $1-\varepsilon$. It is easy to see that the smaller the value of ε the smaller is Φ^*, given the fixed probability p_A.
Figure 4.3 shows the same classification problem of the previous figure but, respectively for (4.3a) and (4.3b), with the Gaussian kernel (parameter 2.5) and the polynomial (degree 2) kernel. What is worth noting is that the sets, as ε varies, are all boxed together, sharing the same behavior as the exact PSR shown in Figure 4.1. Indeed, scaling offset c by the value of ε correctly identifies different safety regions with the desired confidence value: boundary shapes with shades closer to blue result in more conservative safety regions (low values for ε), and conversely shades closer to yellow are for regions of lower confidence (high values for ε).

4.3.2 Hard case

Other experiments were done, considering

- other exponential probability distributions different from Gaussian:
  - *Chi-square distribution*:

$$f(x|\nu) = \frac{x^{(\nu-2)/2} e^{-x/2}}{2^{\frac{\nu}{2}} \Gamma(\nu/2)}$$

    where ν is the degrees of freedom and $\Gamma(\cdot)$ is the Gamma function;
  - *Gamma distribution* :

$$f(x|a,b) = \frac{1}{b^a \Gamma(a)} x^{a-1} e^{\frac{-x}{b}}$$

    where a and b are, respectively, the shape and scale parameters;

- different values of p_A;
- different values of ε;
- different kernels;
- higher dimensions for the samples.

In this case we want to test the effectiveness of our SSVD to resist unbalanced classification: Table 4.1 reports different values of the probability for class A, making explicit the fact that for low values of p_A class A is undersampled and, vice versa, for high values of p_A it is class B that is undersampled.
The table shows the values of $\mathbb{P}_\Omega(A|\Phi^*)$ for different configurations of probability density functions, different sample extraction probabilities and different thresholds for confidence, namely $p_A \in \{0.2, 0.4, 0.5, 0.6, 0.8\}$ and $\varepsilon \in \{0.1, 0.05, 0.01\}$, and the dimension varying from $d = 5$ to $d = 25$ by 5. Regarding the parameters of the exponential distributions, they were set randomly in the following values, respectively for each class:

- Gaussian distribution
  $\boldsymbol{\mu}_A, \boldsymbol{\mu}_B \in \mathbb{R}^d$ with $\mu_{A_i}, \mu_{B_i} \in [-10, 10]$
  $\forall i = 1, \cdots, d$,
  $\sigma_A = \sigma_B = 1$ constant in each direction.
- Chi-squared distribution
  $\nu_A, \nu_B \in \mathbb{N}^d$ with $\nu_{A_i} \in \{1, \dots 60\}$ and $\nu_{B_i} \in \{40, \dots 100\}$, $\forall i = 1, \dots, d$.
- Gamma distribution
  $a_A, b_A, a_B, b_B \in \mathbb{R}^d$ with $a_{A_i} \in [1, 10]$, $b_{A_i} \in [1, 5]$, $a_{B_i} \in [10, 25]$ and $b_{B_i} \in [10, 25]$, $\forall i = 1, \dots, d$.

Polynomial kernel, degree 2, has been used.
From Table 4.1 it can be seen that in almost all tabulated cases the probability value meets the threshold on confidence: the method is more stable for high values of p_A and higher dimensions but there is no evidence of different behaviours among the three different probability distributions, i.e. the points are correctly classified with the desired confidence level for each exponential distribution tested. Moreover, considering the variation of p_A, we can say that the method is robust against unbalanced datasets (small values of p_A mean undersampling of class A).

4.3.3 Comparisons with classic SVM

SSVM is a variant of classical SVM and differs essentially in two respects:

- the hyperplane normal vector $\mathbf{w}$ is the result of weighting the missclassification error in different way;
- the offset is calculated through scaling without using support vectors.

		p_A	0.2			0.4			0.5			0.6			0.8		
		ε	0.1	0.05	0.01	0.1	0.05	0.01	0.1	0.05	0.01	0.1	0.05	0.01	0.1	0.05	0.01
Gauss	Dimension	5	0.90	0.93	0.98	0.85	0.92	1.00	1.00	1.00	1.00	0.9	0.96	1.00	0.90	0.95	1.00
		10	0.89	0.87	0.99	0.89	0.92	0.98	0.92	0.93	0.94	0.90	0.95	0.98	0.91	0.94	0.98
		15	1.00	1.00	1.00	0.91	1.00	1.00	0.91	0.94	1.00	0.90	0.94	0.98	0.90	0.95	0.90
		20	0.84	0.89	0.97	1.00	1.00	1.00	0.91	0.93	1.00	0.89	0.95	0.97	1.00	1.00	1.00
		25	0.98	0.98	0.98	0.89	1.00	1.00	0.88	0.90	1.00	0.94	0.94	0.98	1.00	1.00	1.00
Chi^2	Dimension	5	0.90	0.92	0.96	0.88	0.91	0.98	0.89	0.91	0.95	0.90	0.94	0.97	0.92	0.93	0.99
		10	0.90	0.95	1.00	0.90	0.91	0.95	0.88	0.90	0.93	0.87	0.90	0.97	0.90	0.91	0.98
		15	0.86	0.87	0.92	0.89	0.92	0.99	0.90	0.95	0.97	0.87	0.89	0.93	0.87	0.92	0.97
		20	0.89	0.93	0.98	0.86	0.88	0.90	0.91	0.91	0.93	0.90	0.94	0.98	1.00	1.00	1.00
		25	0.88	0.92	0.95	0.86	0.88	0.89	0.90	0.95	0.99	0.90	0.93	0.96	1.00	1.00	1.00
Gamma	Dimension	5	0.85	0.88	0.92	0.89	0.91	0.93	0.88	0.89	0.91	0.90	0.90	0.93	0.89	0.95	1.00
		10	0.86	0.88	0.94	0.89	0.92	0.93	0.89	0.90	0.90	0.86	0.89	0.92	0.87	0.91	0.96
		15	0.80	0.90	0.94	0.88	0.89	0.90	0.88	0.92	0.93	0.88	0.92	1.00	0.90	0.93	1.00
		20	0.75	0.86	0.93	0.86	0.93	0.94	0.88	0.91	0.92	0.90	0.92	0.96	0.91	1.00	1.00
		25	0.87	0.94	0.99	0.90	0.90	0.90	0.90	0.92	0.97	0.89	0.94	0.96	1.00	1.00	1.00

Table 4.1: The table shows the values of $P_\Omega(A|\Phi^*)$ tabulated for varying values of p_A, ε and probability density functions. The polynomial kernel, degree 2, has been used. It is worth to note that almost all the probability almost all values meet the confidence threshold set, with some minor variation that can be attributed to variability in the dataset.

Therefore, the SSVM can be interpreted as a robust variant of the SVM that can also give assurance about the confidence level of the output.
Moreover, although the SSVM is effectively a classifier, our approach has been developed to approximate the concept of PSR, i.e. defining a region in the input space where we have a sufficiently high level of confidence in predicting a certain label: as a matter of fact, we compared the two methods in predicting the probability of belonging to class A in the case where the data come from samplings of different probabilities.
For this experiment:

- we built a sequence of 100 50-dimensional training sets X_{tr_i} with increasing size from 500 to 5000 points sampled 50 at time from a Gaussian distribution with means $\boldsymbol{\mu}_A$=$[+1, +1, \ldots, +1, +1]$ and $\boldsymbol{\mu}_B$=$[-1, -1, \ldots, -1, -1]$ and variances $\sigma_A = 1$ and $\sigma_B = \frac{1}{2}$ in all directions and 10 different probabilities of observing the class A, $p_A \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.5, 0.6, 0.7, 0.8, 0.9\}$;
- we set for each X_{tr_i}, $i = 1, \ldots, 100$
  - $\boldsymbol{\tau} = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$;
  - $\eta_i = \frac{1}{n_i}$, where n_i is the number of points of X_{tr_i};
  - $\varepsilon = 0.1$ and $\delta = 10^{-3}$;
  - polynomial kernel with degree 2;
- from the training sets we extracted a validation set X_{vl_i}, with dimension computed according to (4.12) for scaling the offset c_i;

- we built a test sets X_{ts} of 100000 sampled in the same way of the training set for evaluating the empirical probability $\mathbb{P}_\Omega(A|\Phi^*)$;
- we trained an optimized SVM on X_{tr_i} and we evaluated it on X_{ts}, $\forall i = 1, \ldots, 100$.

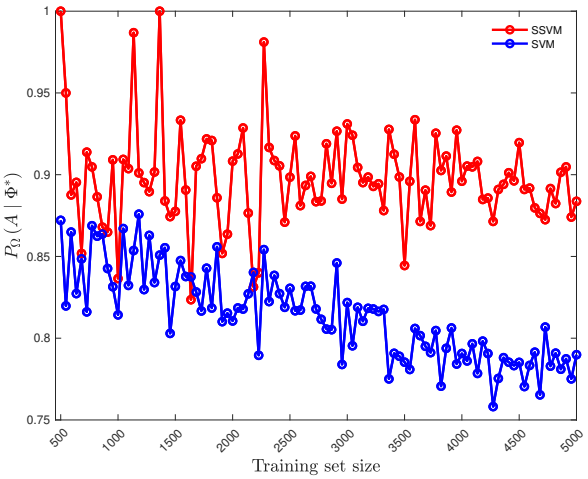


Figure 4.4: Comparison of SSVM and SVM in computing the probability of observing class A by giving the classifier Φ^*.

The Figure 4.4 shows clearly the results of our experiments: the SSVM achieves almost in all training sets its goal to guarantee the confidence of being in the class A at 90% (in average the confidence is 0.9079), overcoming always the classic SVM (which has in average a confidence of 0.8143).
It is worth remarking that the goal of the method, the SSVM, is not to classify the data but to give probabilistic guarantees on the prediction. So what is worth to observe in this experiment is that for obtaining the desired value of confidence on the prediction it is only necessary to tune one parameter, but without retraining the model.

Chapter 5

Rule-based Conformal Safety Regions

In my research, I also proposed a new methodology to link conformal prediction with explainable machine learning by defining a new score function for rule-based models that leverages both rule predictive ability and points geometrical position within rule boundaries. Moreover in this work it is also addressed the problem of defining regions in the feature space where conformal guarantees are satisfied by exploiting techniques to control the number of non-conformal samples in conformal regions based on support vector data description. The overall methodology has been tested with promising results on benchmarks and real datasets, such as vehicle platooning and the prediction of cardiovascular disease.

5.1 Introduction

Combining CP framework with XAI is essential for building a truly trustworthy AI. However, this topic is little explored in current literature, hence this study attempts to address such a research gap through the following contributions:

- A new score function for conformal prediction is defined. This allows to build conformal predictors for rule-based models, by leveraging the combination of the global performance properties of decision rules (i.e., their covering and error) and the geometrical position of the points inside rule boundaries.

- The concept of *conformal critical set*, i.e., the set of target points for which the score function indicates high probabilistic guarantees of the underlying ML model. Moreover, by exploiting SVDD-based techniques for the false positives control, we individuate *conformal critical regions* characterized by the largest number of target points and the minimum non-target points, thus ensuring further precision of the decision-making algorithm.

5.1.1 Notation

Before going into the design details, let us briefly describe the main characteristics and notation of rule-based models.
Let us consider an input example space for classification $\mathcal{T} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{N} \in \mathcal{X} \times \mathcal{Y}$, with $\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^D$ and $y \in \{0, 1\}$.
A rule-based binary classifier $g : \mathcal{X} \longrightarrow \mathcal{Y}$ is expressed by a set of decision rules $\mathcal{R} = \{r_k\}_{k=1}^{M_r}$ in the following form: **if** *premise* **then** *consequence*. The *premise* constitutes the antecedent of the rule and is a logical conjunction ($\wedge$) of conditions c_{i_k}, with $i_k = 1_k, \ldots, N_k$.
Any condition c_{i_k} corresponds to one of the following intervals:

1. $x_{\pi(i)} \geq l_{i_k}$
2. $x_{\pi(i)} \leq u_{i_k}$
3. $l_{i_k} \leq x_{\pi(i)} \leq u_{i_k}$

where l_{i_k}, u_{i_k} are proper numerical thresholds determined by the learning algorithm and $\pi : \mathbb{N} \longrightarrow \mathbb{N}$ denotes the permutation of the indexes of the feature vector $\boldsymbol{x}$ that associates the rule i_{th} condition with the corresponding feature component. Finally, the *consequence* expresses the output class of the decision rule.

Another useful concept in rule-based learning is the notion of *rule relevance*, assigning to each rule a value in the [0,1] range which resembles its predictive ability. Specifically, it is computed by combining the covering $C(r_k)$ and error $E(r_k)$ metrics (commonly known as True Positive Rate and False Positive Rate of the rule, respectively), defined as follows:

$$C(r_k) = \frac{TP(r_k)}{TP(r_k) + FN(r_k)} \tag{5.1}$$

$$E(r_k) = \frac{FP(r_k)}{TN(r_k) + FP(r_k)} \tag{5.2}$$

Denoting with $\hat{y}_j$ the class label predicted by the rule r_k for point $(\boldsymbol{x}_j, y_j)$, $TP(r_k)$ and $FP(r_k)$ are defined as the number of instances that correctly and wrongly satisfy rule r_k, being $\hat{y}_j = y_j$ and $\hat{y}_j \neq y_j$ respectively; conversely, $TN(r_k)$ and $FN(r_k)$ represent the number of samples $(\boldsymbol{x}_j, y_j)$ which do not meet at least one condition in rule r_k, with $\hat{y}_j \neq y_j$ and $\hat{y}_j = y_j$, respectively.
Then, *rule relevance* $R(r_k)$ of rule r_k can be found as:

$$R(r_k) = C(r_k) \cdot (1 - E(r_k)) \tag{5.3}$$

5.2 Rule-Based Conformity

In the conformal prediction framework [132] any score function value $s(\boldsymbol{x}, y)$ is higher for any label y that is less likely to be the correct prediction for the considered point $\boldsymbol{x}$. In this work, the aim was to designing a new score function suitable for rule-based machine learning models.
Considering a generic rule r_k generated by a rule-based model after training, and predicting any output class y, its decision boundary outlines a hyper-rectangle in the feature space (being defined by the premise of the rule). Thus, the closer a point covered by r_k is to this boundary, the higher is its probability of being wrongly covered by the rule. Conversely, points lying inside the rule hyper-rectangle, but farther from the boundary are most probably well conforming to the rule output. So we have to take into account a score that penalizes more the points closer to the classification boundary. For this reason, we introduce the quantity $\gamma = \gamma(\boldsymbol{x}, r_k)$ defined as:

$$\gamma = \sum_{i=1}^{N_k} \left(\frac{1}{d_i^-(\boldsymbol{x}, c_{i_k})} + \frac{1}{d_i^+(\boldsymbol{x}, c_{i_k})} \right), \tag{5.4}$$

where

$$d_i^-(\boldsymbol{x}, c_{i_k}) = |x_{\pi(i)} - l_{i_k}| \quad \text{and} \quad d_i^+(\boldsymbol{x}, c_{i_k}) = |x_{\pi(i)} - u_{i_k}|$$

In order to compute both d_i^- and d_i^+ when either l_{i_k} or u_{i_k} are missing, i.e., when condition c_{i_k} assumes, respectively, the second or the first form described in Section 5.1.1, the minimum and maximum value of feature $x_{\pi(i)}$ across the dataset is considered.

Finally, we normalize γ so that its values vary in the $[0, 1]$ range, thus defining the following parameter:

$$\tau(\boldsymbol{x}, r_k) = \frac{\gamma - \gamma_{\min}}{\gamma_{\max} - \gamma_{\min}} \tag{5.5}$$

This quantity is used in combination with rule relevance to define a score for point $\boldsymbol{x}$ and class label y:

$$s(\boldsymbol{x}, y) \doteq \sum_{r_k \in \mathcal{R}_{\boldsymbol{x}}^y} \tau(\boldsymbol{x}, r_k)(1 - R(r_k)), \tag{5.6}$$

where the sum is on the set $\mathcal{R}_{\boldsymbol{x}}^y$ of rules predicting label y and verified by the input point $\boldsymbol{x}$.

Remark 5.1. The presence of the sum term brings the assumption that multiple rules can *overlap*. However, the proposed score function does not lose generality and remains valid even for models resulting in non-overlapping rules: in this case, ruleset $\mathcal{R}_{\boldsymbol{x}}^y$ will have cardinality fixed to 1.

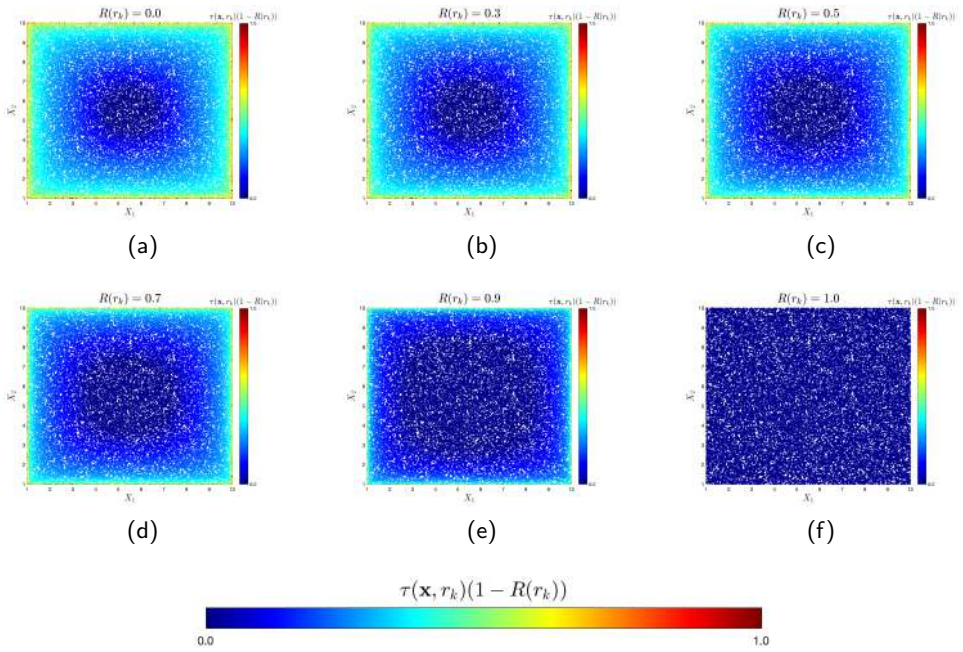


Figure 5.1: Toy example showing rule relevance contribution to the score function

In this way, the introduced score takes into account both the geometrical position of points with respect to rule boundaries and, by depending on rule relevance, the predictive ability of the rules. The latter contribution is expressed through the term $(1 - R(r_k))$ (and not directly through $R(r_k)$) in order to keep the score low when classification has better performance, that is when rule relevance is higher. To better show this behavior, an illustrative example is shown in the next Section 5.2.1.

5.2.1 Toy Examples in 2D

To point out the contribution of rule relevance on the score values, we designed a simple yet explicative example.
Let us consider a bidimensional feature space formed by features X_1 and X_2, and suppose that a rule r_k is learned on such a space, being characterized by the following premise:

$$1 \leq X_1 \leq 10 \ \wedge \ 1 \leq X_2 \leq 10$$

Assuming these thresholds fixed, the geometrical boundaries of the rule remain unchanged and Figure 5.1 shows the effect of increasing relevance values of r_k (from 0 in Fig. 5.1a to 1 in Fig. 5.1f). By looking at the figure, we can observe that when $R(r_k) \leq 0.5$, the score values mainly depend on the geometrical contribution defined by Eq. 5.4 and 5.5: indeed, points that are closer to rule boundaries are well distinguishable to the others. Conversely, as relevance grows ($R(r_k) = 0.7$), its contribution gets more significant, by lowering the score value even for points

that lie close to the boundaries. This is even more pronounced in the extreme case of $R(r_k) = 1$, where the predictive ability of the rule would be so high that it overwhelms the geometrical contribution.
In practice, this design choice handles the possible case when multiple rules have the same geometrical shape (in terms of aspect ratio of their boundary), but different relevance value. As shown in Fig. 5.2, two points (red cross) located at the same distance to the respective rule boundary are scored with a higher value when the rule has a low relevance (left rectangle), and, viceversa, a lower value when the relevance is high (right rectangle).

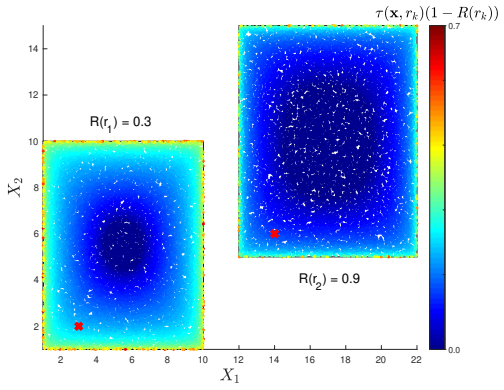


Figure 5.2: Toy example showing two rules r_k, $k = \{1, 2\}$ with relevance $R(r_1) = 0.3$ and $R(r_2) = 0.9$, respectively, whose boundaries share the same aspect ratio. The red cross point in r_1 has a higher score than the one in r_2.

5.3 Experimental Results

In this Section, we present the results of the experiments devoted to test CONFIDERAI score functions, both in terms of canonical metrics in conformal prediction evaluation (i.e., accuracy and efficiency, see Sec. 5.3.2) and of our newly introduced conformal safety set (Sec. 5.3.3).

5.3.1 Datasets description

To evaluate the goodness of CONFIDERAI, we tested the method on 10 datasets, which we briefly describe:

- **P2P** and **SSH**: two datasets concerning peer-to-peer (P2P) and secure shell (SSH) applications of a Domain Name Server (DNS) tunneling detection system [54]; the aim is to detect the presence or absence of DNS attacks by monitoring network traffic and collecting statistical information.

- **BSS**: the Body Signals of Smoking dataset [1] collects personal and biological measurements from subjects, with the aim of predicting if these quantities can represent biomarkers of *smoking* or *non-smoking* habits.
- **CHD**: the Cardiovascular Heart Disease dataset[2] contains patients records with personal, clinical and behavioral features to predict the presence or the absence of a cardiovascular disease.
- **Vehicle Platooning**: the dataset consists of simulations of a vehicle platooning system [93] with a binary output of *collision* or *not-collision* under physical features like the number of cars per platoon or the initial distance between cars.
- **RUL**: the Turbofan Engine Degradation Simulation dataset[3] deals with damage propagation modeling for aircraft engines. The goal is to understand which conditions are inherent to imminent faults of the engine by estimating its Remaining Useful Life.
- **EEG**: the Eye State Classification EEG dataset[4] reports the state of patients' eyes (open or closed) based on continuous electroencephalogram (EEG) measurements.
- **MQTTset** [**mqttset**]: based on Message Queue Telemetry Transportation communication protocol, this dataset collects measurements from different Internet of Things devices to simulate a smart environment; cyber-attacked data are also included to detect *malicious* and *legitimate* traffic.
- **Magic**: the Magic Gamma Telescope dataset[5] reports Monte Carlo simulations of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope to distinguish between gamma and hadron radiation.
- **Fire Alarm**: this dataset[6] contains data to develop an AI-based smoke detection device.

5.3.2 Accuracy and Efficiency

For the evaluation, we considered both accuracy and efficiency, by setting $\varepsilon = 0.01$, $\varepsilon = 0.05$, $\varepsilon = 0.1$ and $\varepsilon = 0.2$. Accuracy was measured by the average error, over the

[1]Reference link: `https://www.kaggle.com/datasets/kukuroo3/body-signal-of-smoking?select=smoking.csv`

[2]Reference link: `https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset`.

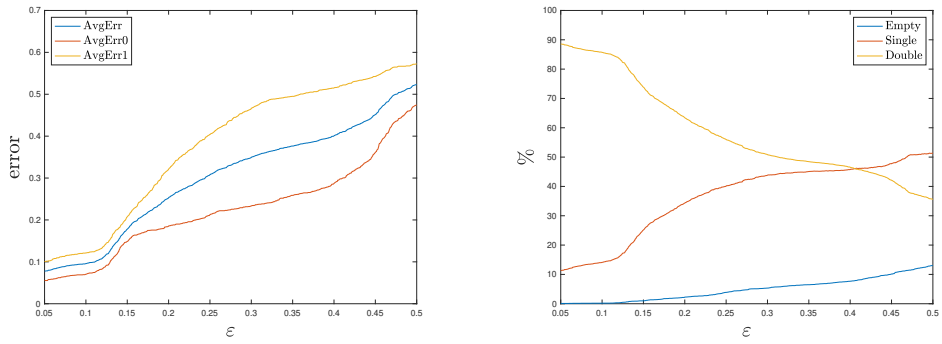
[3]Reference link: `https://www.kaggle.com/datasets/behrad3d/nasa-cmaps`.

[4]Reference link: `https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State`.

[5]Reference link: `https://www.kaggle.com/datasets/abhinand05/magic-gamma-telescope-dataset`.

[6]Reference link: `https://www.kaggle.com/datasets/deepcontractor/smoke-detection-dataset`.

test set, of the conformal prediction sets considering points of both classes (*AvgErr*), only class $y = 0$ points (*AvgErr0*) and only class $y = 1$ points (*AvgErr1*). We remind that an error occurs whenever the true label is not contained in the prediction set. Efficiency was quantified through the percentage of test points prediction sets with singleton predictions (*Single*), no predictions (*Empty*) and two predictions (*Double*). The obtained results are reported in Table 5.1.
The overall metrics computed on the benchmark datasets outline the expected behavior of the conformal prediction. For small values of ε $(0.01, 0.05)$, the average error is always bounded by ε, except for the RUL and Magic datasets, which provide lower results than expected, probably due to the complexity of the dataset. In general, however, the average error increases linearly with ε. As for the size of the conformal set, it varies with ε as should be expected: for small values of ε the model produces more double-sized regions, since in this way it would be "almost certain" that the true label is contained in the conformal set. Then it reduces by increasing ε, allowing the presence of more empty or singleton conformal sets.
Since CONFIDERAI was found out to best perform on the SSH case, we chose this dataset to show the average errors and prediction regions size obtained by varying $\varepsilon \in [0.05, 0.5]$. Figure 5.3 reports the trends of these metrics, pointing out the aforementioned behaviors at the increase of ε. The average error on class 0 , i.e. the *legitimate* samples, is lower than the average error on class 1, i.e. the *attack* points. This is especially evident for $\varepsilon \in [0.2, 0.4]$. Concerning the size, we can notice that for $\varepsilon = 0.4$ the singleton and double-size prediction regions occur in the same percentage (around 45%), while empty predictions keep below 10%.



(a) Average error on both and single classes

(b) Percentage of empty, single-label and two-labels prediction regions

Figure 5.3: Trend of the performance metrics obtained on the SSH dataset by varying $\varepsilon \in [0.05,\ 0.5]$

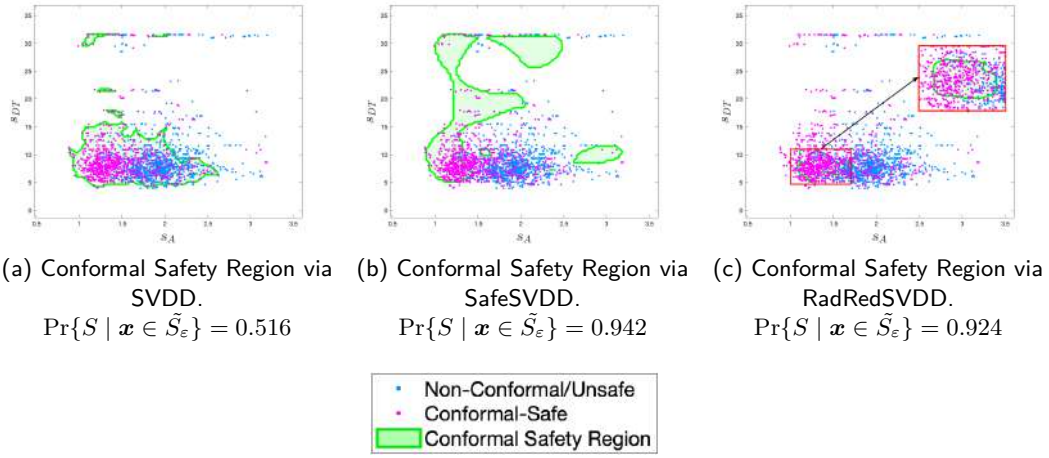
5.3.3 Conformal Safety Sets and Regions

Besides evaluating the error and the size of the obtained prediction regions a Conformal Safety Set at a fixed ε can be identified. Subsequently, test points belonging

Table 5.1: Evaluation metrics for conformal Logic Learning Machine tested on benchmark datasets.

		Error			**Size**			**Conformal Safety Region**		
		AvgErr0	AvgErr1	AvgErr	Empty	Single	Double	Pr_{SVDD}	Pr_{SSVDD}	$\text{Pr}_{\text{RRSVDD}}$
P2P	$\varepsilon = 0.01$	0.018	0.021	0.019	0	0.028	0.972	0.008	0.563	0.019
	$\varepsilon = 0.05$	0.084	0.673	0.377	0	0.386	0.614	0.05	1	0.061
	$\varepsilon = 0.1$	0.176	0.919	0.546	0	0.554	0.446	0.103	0.996	1
	$\varepsilon = 0.2$	0.373	0.922	0.647	0	0.655	0.345	0.2	0.996	1
SSH	$\varepsilon = 0.01$	0.017	0.014	0.015	0	0.023	0.977	0	1	0.125
	$\varepsilon = 0.05$	0.055	0.101	0.078	0.001	0.114	0.885	0.516	0.942	0.924
	$\varepsilon = 0.1$	0.071	0.121	0.096	0.002	0.141	0.857	0.31	1	0.122
	$\varepsilon = 0.2$	0.186	0.321	0.253	0.023	0.343	0.634	0.104	1	0.156
BSS	$\varepsilon = 0.01$	0.022	0.004	0.016	0	0.035	0.965	0.9	0.933	0.6
	$\varepsilon = 0.05$	0.074	0.025	0.057	0.003	0.102	0.895	0.45	0.463	0.07
	$\varepsilon = 0.1$	0.151	0.059	0.118	0.012	0.209	0.779	0.34	0.502	0.56
	$\varepsilon = 0.2$	0.272	0.158	0.231	0.054	0.368	0.578	0.93	0.661	0.135
CHD	$\varepsilon = 0.01$	0.005	0.041	0.024	0.001	0.039	0.96	0	1	1
	$\varepsilon = 0.05$	0.034	0.106	0.072	0.005	0.118	0.877	0	1	0
	$\varepsilon = 0.1$	0.035	0.282	0.164	0.01	0.22	0.77	0.467	0.939	1
	$\varepsilon = 0.2$	0.121	0.363	0.248	0.039	0.353	0.608	0.295	1	1
Vehicle Platooning	$\varepsilon = 0.01$	0.041	0.035	0.038	0	0.064	0.936	0.44	0.603	0.46
	$\varepsilon = 0.05$	0.151	0.126	0.14	0.006	0.208	0.786	0.105	0.697	0.127
	$\varepsilon = 0.1$	0.256	0.205	0.234	0.022	0.326	0.652	0.196	0.744	0.191
	$\varepsilon = 0.2$	0.453	0.344	0.404	0.085	0.469	0.446	0.278	0.99	0.298
RUL	$\varepsilon = 0.01$	0.094	0.083	0.091	0.004	0.168	0.828	0.94	0.418	0.109
	$\varepsilon = 0.05$	0.365	0.281	0.339	0.09	0.465	0.445	0.667	0.268	0.63
	$\varepsilon = 0.1$	0.512	0.383	0.472	0.188	0.493	0.319	0.6	0.328	0.286
	$\varepsilon = 0.2$	0.674	0.474	0.612	0.291	0.509	0.2	0.675	0.365	0.189
EEG	$\varepsilon = 0.01$	0.035	0.045	0.04	0	0.065	0.935	0	1	0
	$\varepsilon = 0.05$	0.098	0.097	0.097	0.007	0.167	0.826	0.63	0.968	1
	$\varepsilon = 0.1$	0.179	0.168	0.174	0.027	0.246	0.728	0.286	1	1
	$\varepsilon = 0.2$	0.315	0.31	0.312	0.072	0.382	0.546	0.189	1	1
MQTTset	$\varepsilon = 0.01$	0	0.011	0.006	0	0.02	0.98	0.257	0.635	0.315
	$\varepsilon = 0.05$	0.001	0.104	0.053	0	0.069	0.931	0.296	0.6	0.316
	$\varepsilon = 0.1$	0.002	0.11	0.057	0	0.073	0.927	0.278	0.603	0.309
	$\varepsilon = 0.2$	0.004	0.391	0.2	0.001	0.216	0.783	0.275	0.574	0.328
Magic	$\varepsilon = 0.01$	0.104	0.141	0.118	0.014	0.198	0.789	0.81	0.419	0.92
	$\varepsilon = 0.05$	0.312	0.497	0.381	0.122	0.494	0.384	0.458	1	1
	$\varepsilon = 0.1$	0.494	0.65	0.552	0.285	0.512	0.203	0.446	0.982	1
	$\varepsilon = 0.2$	0.66	0.766	0.699	0.449	0.456	0.095	0.486	0.983	1
Fire Alarm	$\varepsilon = 0.01$	0.024	0.014	0.019	0	0.02	0.98	0.522	0.798	1
	$\varepsilon = 0.05$	0.315	0.022	0.166	0.005	0.167	0.828	0.495	0.895	1
	$\varepsilon = 0.1$	0.459	0.022	0.237	0.007	0.245	0.748	0.479	0.977	1
	$\varepsilon = 0.2$	0.629	0.022	0.321	0.008	0.332	0.66	0.527	0.873	1

to this set can be labelled as *conformal-safe*, providing a new way to look at the dataset. Indeed, we can train a new classifier to individuate the widest region of only *conformal-safe* points as possible, i.e., a *Conformal Safety Region* (CSR), that is a good approximation of the CSS.
The conformal-safe points enclosed in these regions thus constitute the set of points with label +1 for which the classification is statistically validated. The identification of their boundaries proves very important in real applications, since going outside of them identifies a zone in the feature space where the correct classification of +1 points is no more guaranteed, hence other solutions should be sought, such as another training configuration, another model, etc. In light of the Trustworthy AI principle of technical robustness and safety, this result is crucial.
Construction of conformal safety regions is model-agnostic, i.e. it is possible to use any binary classifier that individuates conformal-safe points, obtaining a region $\tilde{S}_\varepsilon$ that approximates the CSS S_ε. However, it should be pointed out that our target is to construct closed and well defined sets. In this perspective, a good model is the Support Vector Data Description [10], a variation of the well-known SVM, since it is able to define closed envelopes enclosing target points (i.e. conformal-safe) controllable by a radius and a center. In this case, a gaussian-kernel based SVDD has been trained to separate and characterize the conformal-safe points. Moreover, techniques to minimize the number of misclassified points inside the conformal safety region have been adopted as in [136, 171] either by i) performing successive iterations of SVDD inside the classification boundary (SafeSVDD) or ii) reducing the radius of the SVDD (RadRedSVDD) until a predefined threshold on the error is reached. In this case, since conformal safety regions must guarantee the highest level of confidence as possible, the number of false positives (i.e. unsafe or not-conformal points



(a) Conformal Safety Region via SVDD.
$\Pr\{S \mid \boldsymbol{x} \in \tilde{S}_\varepsilon\} = 0.516$

(b) Conformal Safety Region via SafeSVDD.
$\Pr\{S \mid \boldsymbol{x} \in \tilde{S}_\varepsilon\} = 0.942$

(c) Conformal Safety Region via RadRedSVDD.
$\Pr\{S \mid \boldsymbol{x} \in \tilde{S}_\varepsilon\} = 0.924$

Figure 5.4: Conformal safety regions with the (optimized) classical SVDD (5.4a) and the regions obtained reducing the number of non-conformal points with Safe-SVDD (5.4b) and with the reduction of the SVDD radius (5.4c).

wrongly classified as conformal-safe points) has been minimized. We also remark that this kind of control is equivalent to ensuring that the minimum number of non-target points remains within the CSR. The performance for the CSRs is evaluated considering the number of conformal-safe points inside the regions, i.e. the empirical probability $\Pr\{S \mid \boldsymbol{x} \in \tilde{S}_\varepsilon\}$.

Remark 5.2. What we obtain with the CSR $\tilde{S}_\varepsilon$ is only a (good) approximation of the conformal safety set, i.e. a region where conformity is expected but not guaranteed.

The results are shown in Figure 5.4 for SSH dataset at $\varepsilon = 0.05$ and in Table 5.1 for all the other datasets. The values of $\Pr_{\text{SSVDD}}$ and $\Pr_{\text{SSVDD}}$ for the SSH case indicate that within the region the prediction of DNS tunneling attacks is performed correctly in over the 92% of cases, with either one of the methods.

Part II

Counterfactual eXplanations and Rule-based models

Chapter 6

Counterfactual eXplanations

Nowadays, only prediction performances are not anymore sufficient to consider a machine learning algorithm reliable and controllable over the event being predicted. A machine learning algorithm should be considered reliable in the way it allows to extract more knowledge and information than just having a prediction at hand. In this perspective, the counterfactual theory plays a central role. By definition, a counterfactual is the smallest variation of the input such that it changes the predicted behaviour.
In my research, I addressed counterfactuals through Support Vector Data Description (SVDD), binary and multi-class, empowered by explainability and metric to assess the counterfactual quality. As a matter of fact, to generate counterfactual explanations using SVDD, an optimization problem involving kernels, design parameters and evaluation metrics must be solved. In this sense, SVDD proved to be a good approach for generating counterfactual eXpalanations since it is possible to prove that analytical and exact solution may be found (under relaxed hypothesis, but suggestions can lead to think that a generalization can be found). As usual, after evaluating the method on synthetic data, real world applications were tested. The results obtained are more than encouraging and suggest that the proposed method can compete with those in the state of the art.

6.1 Introduction

Counterfactual explanations (CEs), a concept borrowed from philosophy of language and logic, has been first declined in the context of machine learning by Wachter et al [74] as *the minimal change that is required in the input features of a certain observation in order for the prediction of that observation to fall into the opposite class*, in a binary classification problem. Specifically, a change of a certain delta in the features describing the observation $\boldsymbol{x}$, belonging to class $\mathbf{C}$, leads to the generation of an observation $\boldsymbol{x}'$ (i.e., the counterfactual of $\boldsymbol{x}$) that will be classified as belonging to class $\mathbf{C}'$. These kind of local explanations are assuming a certain importance, especially in machine learning models dealing with images [151], as they allow to add a certain degree of interpretability to the underlying behavior of complex

models like neural networks, in line with the demand of the European General Data Protection Regulation (GDPR)[1] for greater transparency when handling decisions made by a model. Different approaches have been recently proposed to produce realistic and feasible counterfactuals to provide local explanations for automated decision making processes. Below, the bullet-point list provides an overview of related literature with regards to the methods for CEs generation:

- White et al [129]:
  - CLEAR: minimization of the fidelity error, obtained by iteratively comparing progressive b-perturbations of each single feature with estimates of b-perturbations calculated using a local regression equation built around the initial point.
- Poyiadzi et al [123]:
  - FACE: minimization of the f-distance describing the trade-off between path length and data density along the path, through the Shortest Path First Algorithm applied to a graph constructed over data points by using KDE, KNN or ε-graph.
- Van Looveren et al [153]:
  - Addition of a prototype loss term in the objective function, to guide and fasten the search process. Encoders or K-d trees may be used to define class prototypes.
- Mochaourab et al [144]:
  - Bisection method: starting from two prototypes with opposite class, according to Privacy preserving SVM with RBF kernel.
- Dhurandhar et al [82]:
  - CEM: optimization of the perturbation variable using the fast iterative shrinkage-thresholding algorithm (FISTA) coupled with the use of a CAE to evaluate the distance from the data manifold.
- Albini et al [113]:
  - Mapping the variables that influence the assignment of observations to classifications in Bayesian Network classifiers (single or multi-label, binary or multidimensional).

Moreover, White et al [129] determined counterfactuals by applying minimum perturbations for each feature separately and use them to generate local regression models, then evaluating the fidelity of these regressions, in five different case studies.

[1] https://gdpr.eu/tag/gdpr/

Poyiadzi et al [123], instead, proposed a method for generating CEs by considering a trade-off between the length of the path from the point to its corresponding counterfactual and the data density along this path. Finally, Mochaourab et al [144] considered the design of robust CEs for privacy preserving mechanisms based on binary Support Vector Machines, by applying the bisection method between two points belonging to different classes and evaluating the trade off between accuracy, privacy and explainability.

Whether an observation belongs to a certain class may depend on two categories of features: *controllable features*, which can be manipulated through internal/external intervention (e.g., therapies or lifestyle changes in clinical classification problems or control algorithms in systems modelling and control problems) and *non-controllable features*, which by their nature are not manipulable (e.g., the age of a subject in health prediction algorithms). Therefore, the search for realistic counterfactuals should be performed by perturbing only controllable variables. To my and my team knowledge, the only attempt to force the generated CEs to have no change in terms of non-controllable characteristics was carried out by Nemirovsy et al [148] who developed a method to produce counterfactuals able to provide actionable feedbacks in real-time using Generative Adversarial Networks (GANs). However, in that case, feature immutability was imposed after the application of the counterfactual search algorithm by setting the values of non-controllable features to the original values rather than to the values suggested by the counterfactual search algorithm. By contrast, in this study, for the first time, the search for counterfactuals is guided by directly perturbing only controllable features.

Previous related works validated the proposed CEs with respect to explanations obtained with other local explainability methods, like Local Interpretable Model-agnostic Explanations (LIME) or Layer-Wise Relevance Propagation (LRP) [129, 82] or with respect to other state-of-the-art method for generation of CEs [123, 148, 113]. Often, the validation measure relies on verifying that the CE is correctly associated with its target outcome, based on the prediction of a classifier. However, this measure is characterized by a degree of uncertainty, since it is not guaranteed that the real class matches the predicted class. To our knowledge, none of the approaches presented in the literature is supported by a validation of the generated CEs with computational simulations, capable of verifying that the CE belongs to a certain class, and rule-based models that explain the reason for this belonging.

In my research, a novel methodology for counterfactual generation and validation was introduced. The counterfactuals generation method uses regions defined by Two Class-Support Vector Data Descriptors (TC-SVDDs) and was developed in both analytical and numerical form. The validation method combines computational simulations and eXplainable AI (XAI), specifically in the form of rule-based classification of counterfactuals.

As subsequent work, a generalization to the multiclass case was developed. As a matter of fact, examples may help understand the importance of counterfactual reasoning in multi-class situations. In healthcare, several diseases present different

stages of severity (e.g., cancer, chronic obstructive pulmonary disease...) that can worsen drastically in a short time if not properly treated. In this case, multi-class counterfactuals can be a valuable instrument to monitor the stage of disease progression in order to detect minimal changes in the patient's condition and apply appropriate countermeasures before the disease progresses to the next stage. Another example may involve the study of the transitions of a phenomenon that develops over several stages (e.g., A, B, C, D). The counterfactual analysis can be useful to check for differences between different transitions (e.g., direct paths skipping intermediate transitions or progressive sequential paths). Several practical applications may be mentioned of this type, such as vehicular platooning [143] and predictive maintenance [30].
The objective of this extension was to develop a novel method based on Support Vector Data Description under multi-class setting (MC-SVDD) to identify multiple counterfactual explanations from a given observation under varying constraints. The use of SVDD envelopes may provide several advantages, e.g. detection of anomalous points (outside SVDD clusters) and flexible contour of different classes, by including the control of false positives/false negatives rates [163]. To the best of my knowledge, this is the first work aimed at the generation of counterfactuals for multi-class classification problems based on data envelopes extracted via SVDD. The method developed in this study addresses: **1)** explainability, through the use of of counterfactuals, **2)** controllability of counterfactuals via MC-SVDD and **3)** validation of counterfactual quality in terms of availability, actionability, similarity and discriminative power. Tabular open source datasets are used and made available with source code via github[2]with respect to comparison with DICE algorithm). Indeed, the use of SVDD allows data points to be reliably and flexibly contoured. Furthermore, the SVDD allows for the elimination of anomalous points, ensuring that the factual and the corresponding counterfactual explanation are points that are sufficiently representative of the class to which they belong.
Regarding the state of the art around multi-class classification and counterfactuals, multi-class classification is the task of classifying a new instance into one among at least three classes. As always, when the variability of a problem increases, so does the effort to solve it. There exist different approaches to address the increase of the classes. For example, some algorithms, such as decision trees and Neural Networks, automatically handle multiple outputs. Other algorithms provide exclusively binary outputs (e.g., SVM, logistic regression, perceptron). In these cases, binary classifiers must be adapted to handle multiple outputs. Therefore, we can distinguish two types of multi-class classification techniques [91]: *one-vs-one* and *one-vs-rest*. In one-vs-one techniques the problem is divided into $\frac{m(m-1)}{2}$ binary classifiers, where m is the number of classes and each binary classifier predicts a class label. Then, an instance is assigned to the class with the highest number of counts. In one-vs-rest techniques, instead, the model is trained for m different datasets, where each target class is trained against the rest of the classes. Then, an instance is assigned to the class with the highest probability.

[2] `https://github.com/AlbiCarle/MUCH.git`

Due to the their incremental adaptation to multiple outputs, these approaches lack a comprehensive view of relationship in among the classes. In addition, due to multiple trainings, they are not feasible for large datasets. The approach here proposed *Multi-ClassSVDD* (MC-SVDD)[3], solves the problem in one shot, without repetitive adaptations. All uncertainties and data characteristics are handled at the same time, providing a result that best fits the problem [167]. The algorithm generalizes the well-known SVDD by Tax and Duin [11] to the multi-class case, quite naturally as an extension of the original method.

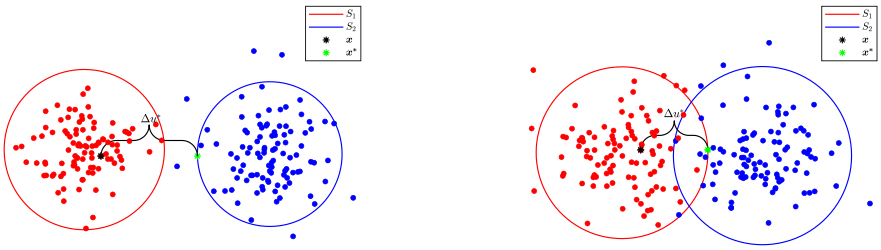
Other attempts address multi-class SVDD, but identifying objects belonging to multiple anomalies rather than providing canonical classification. The algorithm proposed by [45] generalizes the unsupervised one class classifier of [49] to multiple outputs. The multi-class SVDD algorithm proposed in [45] does not consider the fact that the classification regions (i.e., the hyperspheres) may intersect with each other, thus simply defining a generalization of unsupervised one class classification (OOC) [49] in which the algorithm does not consider relationships between classes. A different approach is proposed by [61], in which the canonical SVDD is merged with binary tree to handle the multi classification problems. Guo et al. [140] proposed a multi kernel learning adaptation to SVDD (MKL-SVDD) to design the kernel weights for multiple kernels and obtain the optimal kernel combination. Hou et al. [141] developed a multi-class SVDD algorithm to classify multiple classes of planetary gear faults based on the method proposed by [41] that minimizes the radius of each hypersphere, while maximizing the distance between them. However, the boundary between couples of classes is optimized for each pair of centers, without including further constraints inherent to the other classes. Recently, a generalization of SVDD to the multi-class case has been proposed [126], but the focus is on anomaly detection.

Long story short, whether counterfactual generation relates to a binary classification problem or a multiclass one, the basic idea does not change: counterfactual explanations are local post-hoc XAI techniques, both model-specific and model agnostic, that allow the output of, basically, any ML model to be controlled and modified. In my research, which is still ongoing, I am looking for a general definition of a counterfactual explanation, one that takes into account all the properties discussed so far and allows for connection and comparison with similar but not yet related areas of research in ML, such as adversarial ML or conformal prediction.

6.2 Counterfactual building via Two Class SVDD

Suppose we have a dataset $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^N \times \{-1, +1\}$, $N \geq 2$, consisting of a subset of controllable features $\mathbf{u}$ and a subset of non-controllable features $\mathbf{z}$, so that an observation $\boldsymbol{x} \in \mathcal{X}$ can be described as

[3] `https://github.com/AlbiCarle/MultiClass_SVDD.git`



(a) Counterfactual solution for $S_1 \cap S_2 = \emptyset$. The solution in this case is obtained by simply posing $\lambda_2 = 0$, i.e., imposing nullity on the constraint (6.3c).

(b) Counterfactual solution for $S_1 \cap S_2 \neq \emptyset$. In this case the optimal solution is not on the edge of the region S_2 but it is inside it.

Figure 6.1: Counterfactual solutions for a 2-dimensional linear TC-SVDD with Euclidean distance. Points were sampled from a Gaussian distribution of variance 0.5 and mean 0 and 5 for red and blue points, respectively. The controllable variables lie on the abscissas while those not controllable on the ordinates, i.e. $u\mathrm{O}z$ plane.

$$\boldsymbol{x} = \left(u^1, u^2, \ldots, u^n, z^1, z^2, \ldots, z^m\right) \in R^{n+m=N}$$

We perform a TC-SVDD classification as in [37], obtaining two regions

$$S_1 \doteq \{\boldsymbol{x} \in R^N : \|\boldsymbol{x} - \mathbf{a}_1\|^2 \leq R_1^2, \|\boldsymbol{x} - \mathbf{a}_2\|^2 \geq R_2^2\}$$

and

$$S_2 \doteq \{\boldsymbol{x} \in R^N : \|\boldsymbol{x} - \mathbf{a}_2\|^2 \leq R_2^2, \|\boldsymbol{x} - \mathbf{a}_1\|^2 \geq R_1^2\},$$

where $R_1^2, R_2^2, \mathbf{a}_1, \mathbf{a}_2$ are, respectively, the radii and the centers of the spheres of the computed TC-SVDD.

Given an object $\boldsymbol{x} = (\mathbf{u}, \mathbf{z}) \in S_1$, our goal is to determine the minimum variation $\Delta\mathbf{u}^*$ of the controllable variables so that the point

$$\boldsymbol{x}^* = (\mathbf{u} + \Delta\mathbf{u}^*, \mathbf{z}) \tag{6.1}$$

belongs to the class S_2. To determine $\Delta\mathbf{u}^*$, we define the following minimization problem

$$\min_{\Delta\mathbf{u} \in \mathbb{R}^n} \quad d\big(\boldsymbol{x}, (\mathbf{u} + \Delta\mathbf{u}, \mathbf{z})\big) \tag{6.2a}$$

$$\text{subject to} \quad \|(\mathbf{u} + \Delta\mathbf{u}, \mathbf{z}) - \mathbf{a}_2\|^2 \leq R_2^2 \tag{6.2b}$$

$$\|(\mathbf{u} + \Delta\mathbf{u}, \mathbf{z}) - \mathbf{a}_1\|^2 \geq R_1^2 \tag{6.2c}$$

where d is a distance and (6.2b), (6.2c) are the constraints that require $\boldsymbol{x}^*$ to belong to S_2 and not to S_1, respectively. In other words, the *counterfactual* $\boldsymbol{x}^*$ is the nearest point, with respect to distance d, that belongs to the class opposite to the original class of a given point $\boldsymbol{x}$, taking into account that *only* controllable features $\mathbf{u}$ can be modified.
Finding an analytical solution of (6.2) is not an easy task and might be impossible since the space of constraints is not convex (i.e., the constraint (6.2c) is not convex), also it is necessary to take into account the choice of distance d. However, there are some cases where it is possible to analytically explicate the solution of (6.2), for example choosing as distance the Euclidean norm, performing a linear TC-SVDD and assuming to be only in two dimensions, with one feature controllable and the other non-controllable. In other cases, the solution of (6.2) will be performed numerically by sampling the classification regions with a grid and searching for the closest point of a given observation with respect to a fixed distance.

6.2.1 $\mathbb{R}^2$ analytical solution

Let be $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^2 \times \{-1, 1\}$ a labelled two-dimensional dataset, in which each object $\boldsymbol{x} \in \mathcal{X}$ consists of a controllable component u and a non-controllable one z, i.e. $\boldsymbol{x} = (u, z) \in \mathbb{R}^2$. After performing a linear TC-SVDD [37] and determining two regions $S_1, S_2 \subset \mathbb{R}^2$, our goal is, given an object $\boldsymbol{x} = (u, z) \in S_1$, to find the minimum change in the controllable variable Δu^* so that the object $\boldsymbol{x}^* = (u+\Delta u^*, z)$ is the closest point to $\boldsymbol{x}$ belonging to S_2 and not belonging to S_1.
In $\mathbb{R}^2$, the problem to be solved is the following:

$$\begin{aligned} \min_{\Delta u \in \mathbb{R}} \quad & \|(u, z) - (u + \Delta u, z)\|^2 && \text{(6.3a)} \\ \text{subject to} \quad & \|(u + \Delta u, z) - \mathbf{a}_2\|^2 \leq R_2^2 && \text{(6.3b)} \\ & \|(u + \Delta u, z) - \mathbf{a}_1\|^2 \geq R_1^2 && \text{(6.3c)} \end{aligned}$$

Two slack variables ξ_1, ξ_2 are introduced and the above problem changes in:

$$\begin{aligned} \min_{\Delta u \in \mathbb{R}} \quad & \Delta u^2 + D_1\xi_1 + D_2\xi_2 && \text{(6.4a)} \\ \text{subject to} \quad & \|(u + \Delta u, z) - \mathbf{a}_2\|^2 \leq R_2^2 + \xi_1,\ \xi_1 \geq 0 && \text{(6.4b)} \\ & \|(u + \Delta u, z) - \mathbf{a}_1\|^2 \geq R_1^2 - \xi_2,\ \xi_2 \geq 0 && \text{(6.4c)} \end{aligned}$$

where the parameters D_1, D_2 control the trade-off between the distance and the error.
Introducing the Lagrange multipliers $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$ we get the Lagrangian function

$$\begin{aligned} \mathcal{L}(\Delta u, \xi_1, \xi_2) = \Delta u^2 &+ D_1\xi_1 + D_2\xi_2 \\ &- \lambda_1\Big(R_2^2 + \xi_1 - \|(u + \Delta u, z) - \mathbf{a}_2\|^2\Big) \\ &- \lambda_2\Big(\|(u + \Delta u, z) - \mathbf{a}_1\|^2 - R_1^2 + \xi_2\Big) \\ &- \lambda_3\xi_1 - \lambda_4\xi_2 \end{aligned} \tag{6.5}$$

Setting partial derivatives to zero gives the following constraints:

$$\frac{\partial \mathcal{L}}{\partial \Delta u} = 0 \Rightarrow \Delta u = \frac{\Big(\lambda_2\big(u - a_1^u\big) - \lambda_1\big(u - a_2^u\big)\Big)}{1 + \lambda_1 - \lambda_2} \tag{6.6}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_1} = 0 \Rightarrow D_1 - \lambda_1 - \lambda_3 = 0 \Rightarrow 0 \leq \lambda_1 \leq D_1 \tag{6.7}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_2} = 0 \Rightarrow D_2 - \lambda_2 - \lambda_4 = 0 \Rightarrow 0 \leq \lambda_2 \leq D_2 \tag{6.8}$$

where a_1^u, a_2^u are the projections of $\mathbf{a}_1, \mathbf{a}_2$ onto the controllable variable u.
By substituting (6.6) into the expression of $\mathcal{L}$ we get:

$$\begin{aligned} \mathcal{L}(\lambda_1, \lambda_2) = &- \frac{\Big(\lambda_2\big(u - a_1^u\big) - \lambda_1\big(u - a_2^u\big)\Big)^2}{1 + \lambda_1 - \lambda_2} \\ &- \lambda_1\Big(R_2^2 - \|(u, z) - \mathbf{a}_2\|^2\Big) - \lambda_2\Big(\|(u, z) - \mathbf{a}_1\|^2 - R_1^2\Big) \end{aligned} \tag{6.9}$$

which must be maximized under the constraints (6.7) and (6.8) to get λ_1^* and λ_2^* to be substituted into (6.6) to obtain the minimum variation Δu^*.

6.2.2 Numerical Solution

As the size of the feature space increases and for more complicated distances d or kernels, the solution of (6.2) may be analytically unfeasible. Thus, a grid-search algorithm has been developed.
Algorithm 4 returns the set $\mathcal{C}$ of counterfactuals of points belonging to S_1. Of course, the same procedure can be applied to find the counterfactuals of the points belonging to S_2 simply by reversing the roles of S_1 and S_2. For better understanding, Table 6.2.2 shows the meaning of the symbols and variables used in **Algorithm 4**.

Algorithm 4 `CounterfactualSVDD`
Dataset $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^N \times \{-1, +1\}$ is divided in training set $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$ and validation set $\mathcal{X}_{vl} \times \mathcal{Y}_{vl}$.
A TC-SVDD [37] is performed on $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$ and validated on $\mathcal{X}_{vl} \times \mathcal{Y}_{vl}$ in order to derive S_1 and S_2.
$N_{\text{cntrfctls}} > 0$ is fixed.

---

1. $\mathcal{C} = [\,]$
2. **Sample** uniformly a new dataset G
3. $G_1 \cup G_2 \doteq G \cap \big(S_1 \triangle S_2\big)$
4. **for** $i = 1 : N_{\text{cntrfctls}}$
4.1 $\boldsymbol{x}_i = (\mathbf{u}_i, \mathbf{z}_i) \in S_1$

Line	Symbol	Description
1.	$\mathcal{C}$	Set of counterfactuals
3.	$\triangle$	Symmetric difference, $G_1 = G \cap (S_1 \backslash S_2)$ $G_2 = G \cup (S_2 \backslash S_1)$
4.1	$\boldsymbol{x}_i$	Factual point
4.2	d	Distance function
4.2	$G_{2_{\mathbf{z}=\mathbf{z}_i}}$	G_2 grid points with component $\mathbf{z}$ equal to $\mathbf{z}_i$
4.3	$\boldsymbol{x}'_i$	Counterfactual point

Table 6.1: **Algorithm 4** legend.

4.2 $d_i = d\big(\boldsymbol{x}_i, G_{2_{|\mathbf{z}=\mathbf{z}_\mathbf{i}}}\big)$
4.3 $\boldsymbol{x}'_i = \min(d_i)$
4.4 **if**$\big(\boldsymbol{x}_i \in S_1$ & $\boldsymbol{x}'_i \in S_2\big)$
4.4.1 $\mathcal{C} = \mathcal{C} \cup \{\boldsymbol{x}'_i\}$
4.5 **end**
5. **end**
6. **return** $\mathcal{C}$

The points for which a counterfactual is desired are randomly or directly sampled in S_1, while their counterfactual is sought in the grid G_2, with the non-controllable features fixed. Thus, the accuracy of the counterfactual is related to the granularity of the grid: the denser the grid, the more accurate the counterfactual will be. Moreover, since the concept of counterfactual is closely related to explainability, a set of rules for each TC-SVDD class, $\mathcal{R}(S_i)$, is defined according to `ExplainableSVDD` algorithm [135, 136]. This is a further validation that will then also be used as a basis for extracting knowledge from the rules that characterize counterfactuals: if the point $\boldsymbol{x}_i$ and its counterfactual $\boldsymbol{x}'_i$ belong to the set of rules defining the respective classes then we accept $\boldsymbol{x}'_i$ as the counterfactual of $\boldsymbol{x}_i$. Since the counterfactual determined by the algorithm is an approximation of the real counterfactual, a metric of the quality of the extracted counterfactual is needed. Given a point, its counterfactual is, by definition, the nearest point belonging to the opposite class. Thus, a straightforward metric for evaluating the quality q of the counterfactual $\boldsymbol{x}'$ of a point $\boldsymbol{x} \in S_1$ is to evaluate its distance from S_1:

$$q = d(\mathbf{a}_1, \boldsymbol{x}') - R_1 \tag{6.10}$$

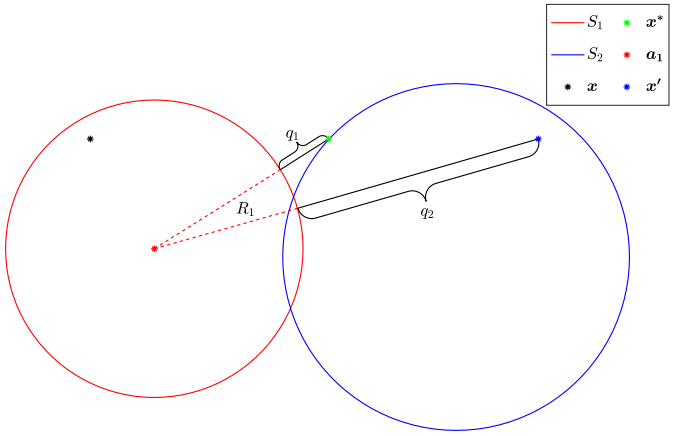


Figure 6.2: 2D-linear example of CQ: this metric evaluates the goodness of the counterfactual, the closer q is to zero the more the counterfactual is optimal in terms of minimum distance. In the figure, $q_2 > q_1$ and the blue counterfactual $\boldsymbol{x}'$ is worst than the green (optimal) one $\boldsymbol{x}^*$.

where $\mathbf{a}_1$ and R_1 are respectively the center and the radius of S_1. We define this new metric as *Counterfactual Quality* (CQ).
From Figure 6.2 it is easy to see that the lower the q, the better the counterfactual and if $q < 0$ then the counterfactual determined is incorrect.

6.3 Clarifying example

The following safety-critical application is considered. Vehicle platooning is one of the most challenging problems in smart mobility scenarios. It consists of a group of vehicles interconnected via wireless that travel autonomously; the aim is to find a compromise between performance (e.g., maximize speed and minimize reciprocal distance, thus minimizing air drag resistance and fuel consumption, too) and safety (avoid collisions, even in the presence of anomalous events, such as sudden brakes or cyberattacks, [146])
The aim here is to determine what is the minimum variation in terms of controllable factors (i.e, the initial mutual distance and speed between two consecutive vehicles in the platoon, respectively d_0 and v_0) that allows for a change in system safety (collision / non-collision or vice versa; a point of the dataset is labelled as collision if the distance between any couples of vehicles, during the simulation run, becomes lower than 2 meters).

6.3.1 Data set Description

#	output	cond1	cond2	cond3	cond4	cond5	covering	error
1	-1	$F \leq 4690$	$d_{ms} \leq 478$	$d_0 > 13$			0.50	0.04
2	-1	$F \leq 4238$	$m \leq 1963$	$10 < d_{ms} \leq 335$	$v_0 \leq 111$		0.49	0.04
3	-1	$106 < F \leq 2287$	$17 < d_{ms} \leq 496$	$7 < d_0 \leq 17$	$v_0 \leq 118$	$p < 0.5$	0.43	0.04
4	-1	$N \leq 6$	$d_{ms} \leq 493$	$d_0 > 13$			0.37	0.04
5	-1	$m > 806$	$d_0 > 14$				0.37	0.04
6	-1	$F \leq 4941$	$566 < m \leq 1990$	$2 < d_{ms} \leq 399$	$v_0 \leq 80$		0.34	0.04
7	-1	$F \leq 1432$	$m > 801$	$d_{ms} \leq 482$			0.27	0.04
8	-1	$2 < d_{ms} \leq 97$	$d_0 > 8$				0.20	0.03
9	-1	$m > 753$	$2 < d_{ms} \leq 72$	$d_0 > 5$			0.17	0.04
10	-1	$N \leq 6$	$1365 < F \leq 4964$	$m > 1586$	$d_{ms} > 77$	$v_0 > 66$	0.09	0.04
11	+1	$F > 1116$	$d_{ms} > 29$	$d_0 \leq 8$			0.52	0.04
12	+1	$d_0 \leq 7$	$v_0 > 79$				0.37	0.03
13	+1	$d_{ms} > 45$	$d_0 \leq 15$	$p \geq 0.58$			0.29	0.04
14	+1	$N \leq 7$	$3714 < F \leq 4606$	$d_0 \leq 16$	$54 < v_0 \leq 118$		0.26	0.05

Table 6.2: Explainable rules extracted from SVDD through the algorithm `ExplainableSVDD` as in [135, 136].

Factuals										Counterfactuals									
$\mathbf{d_0}$	$\mathbf{v_0}$	**N**	**F**	**m**	$\mathbf{d_{ms}}$	**p**	**SVDD**	**LLM**	**Rule**	$\mathbf{d_0}$	$\mathbf{v_0}$	**N**	**F**	**m**	$\mathbf{d_{ms}}$	**p**	**SVDD**	**LLM**	**Rule**	$\boldsymbol{\Delta u^*}$
5	117	8	2976	952	81	0.44	+1	+1	11	18	111	8	2976	952	81	0.44	-1	-1	1	$(13, -6)$
6	97	8	4898	1215	92	0.3	+1	+1	11	15	51	8	4898	1215	92	0.3	-1	-1	5	$(6, -49)$
6	82	4	966	1271	398	0.5	+1	+1	12	6	51	4	966	1271	398	0.5	-1	-1	6	$(0, -31)$
7	73	8	1290	807	338	0.43	+1	+1	11	15	50	8	1290	807	338	0.43	-1	-1	1	$(8, -23)$
5	65	3	1117	535	329	0.48	+1	+1	12	16	116	3	1117	535	329	0.48	-1	-1	1	$(11, 51)$
7	91	6	973	708	458	0.13	+1	+1	12	16	55	6	973	708	458	0.13	-1	-1	1	$(9, -36)$
7	108	5	3451	1895	478	0.19	+1	+1	11	18	84	5	3451	1895	478	0.19	-1	-1	4	$(11, -24)$
8	99	8	3993	634	380	0.01	+1	+1	11	17	99	8	3993	634	380	0.01	-1	-1	5	$(9, 0)$
6	76	6	1785	744	370	0.11	+1	+1	11	18	56	6	1785	744	370	0.11	-1	-1	1	$(12, -20)$
5	119	3	2333	554	272	0.31	+1	+1	11	18	50	3	2333	554	272	0.31	-1	-1	1	$(13, -69)$

Table 6.3: Counterfactual explanation table of ten points randomly sampled from the set of 10000 extracted collision points. The last column contains the minimum change $\boldsymbol{\Delta u}^*$ of the controllable features d_0, initial distance, and v_0, initial velocity, of the platoon.

The data set concerning collision prediction in vehicle platooning is taken from [93, 146][4]. The machine learning solution is based on a supervised classification task that maps the features into a potential collision in the near future; features are: braking force of lead vehicle (at the top of the platoon), current speed, distance and acceleration, number and weight of vehicles, as well as quality of service of the communication channel (loss probability and delay). Controllable variables are speed and distance only, thus making the restrictions on counterfactual generation (with respect to the other variables), as well as the search in the grid of the destination SVDD, very tight.

[4] `https://github.com/mopamopa/Cyberplatooning` and `https://github.com/mopamopa/Platooning`

In this scenario, the counterfactual explanation can play an effective role in improving the safety of the platooning system: given a combination of the platoon input parameters that brings the system into collision, the counterfactual finds the minimal change in the controllable features such that the platoon no longer collides. Finding such a minimal change simplifies the recovery operation (from collision).
The behaviour of the platooning system is synthesised by the following vector of features:

$$\mathbf{I} = [N, F, m, d_{ms}, p, d_0, v_0]$$

where N is the total number of vehicles of the platoon, F is the braking force applied by the leader, m is the weight of the vehicles, d_{ms} is the communication delay in milliseconds, p is the probability of packet loss, and d_0 and v_0 are the mutual distance and speed between each pair of vehicles in the initial condition.
Data points are sampled by implementing the CACC simulator as in [146] in the following ranges:

$$N{\in}[3,8],\ F{\in}[1000,5000]\ \text{N},\ m{\in}[500,2000]\ \text{Kg},$$
$$d_{ms}{\in}[0,1000]\ ms,\ \ d_0{\in}[4,20]\ \text{m},\ v_0{\in}[30,130]\ \text{Km/h},\ \mathbf{p}{\in}[0,1].$$

The considered ranges are very challenging as they cover a very large set of working conditions. As already said, since the control of the dynamical system reacts by changing the initial distance and speed, we consider the variables d_0 and v_0 as the only controllable ones and the others as non-controllable, therefore, named $\mathcal{X}_{PL}$ the platooning dataset, an observation $\boldsymbol{x} \in \mathcal{X}_{PL}$ can be written as

$$\boldsymbol{x} = (\mathbf{u}, \mathbf{z})$$

where $\mathbf{u} = (d_0, v_0)$ and $\mathbf{z} = (N, F, m, d_{ms}, p)$.
The analysed platooning data set includes 20000 records with equally distributed samples for the collision (+1) and non-collision (-1) classes. A TC-SVDD with Gaussian Kernel [136] has been trained ($\sigma = 1.87$, $C_1 = C_2 = 1$, $C_3 = 1/(\nu N_1)$, $C_4 = 1/(\nu N_{-1})$, where N_1 and N_{-1} are the sizes of the collision and non collision class, respectively, and $\nu = 0.05$ as in [37]) on 60% of the data and evaluated on the remaining 40%. A set of 10000 CEs has been generated through the implementation of **Algorithm 4** and validated both with rule-analysis and simulations.
Figure 6.3 presents the scatterplots of all the possible pairs of features in the platooning data set, grouped by target class, and reveals how the separation between safety and collision may be hardly found without complex combinations of more than two features.

6.3.2 Results

The TC-SVDD trained on the platooning data achieved the following classification performance: training accuracy of 0.88, test accuracy of 0.88, sensitivity of 1.00, specificity of 0.75. LLM decision rules describing the two SVDD regions are extracted as in [135, 136] and presented in Table 6.2. Specifically, the collision region is

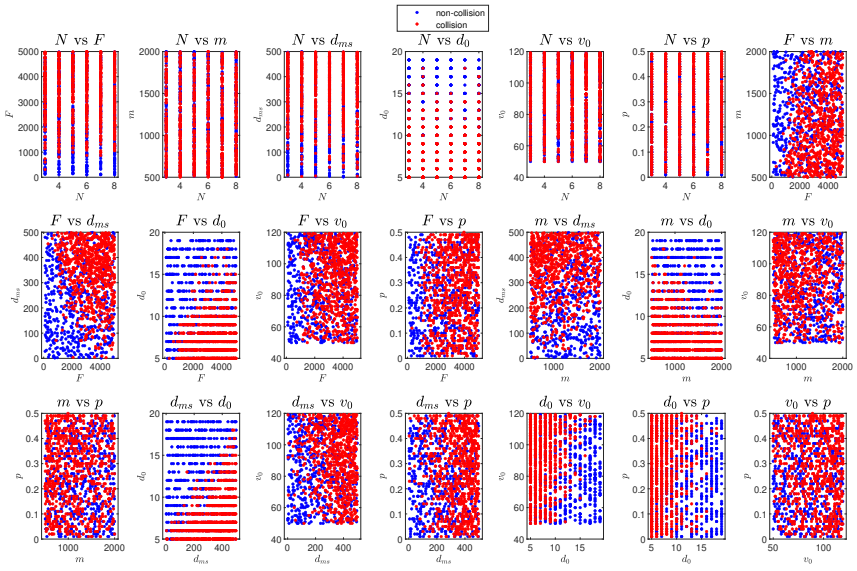
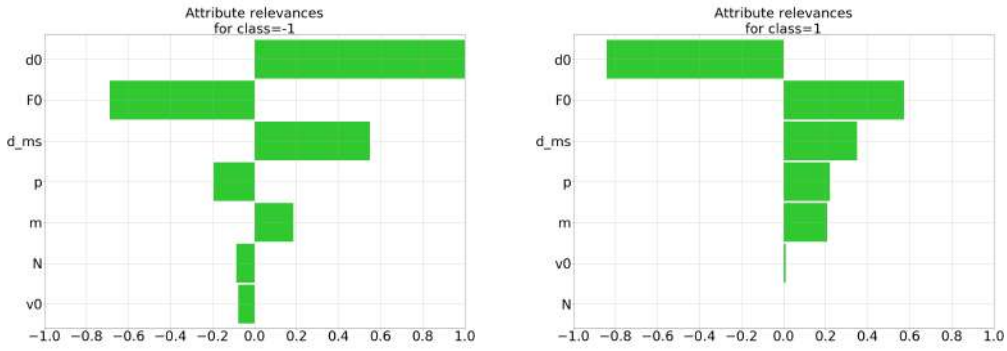


Figure 6.3: Scatter plot of each pair of variables in the platooning data set. Red dots indicate collision and blue dots indicate non-collision.

described by four rules (average number of conditions =2.75), whereas the non collision region is described by ten rules (average number of conditions =3.3).
The feature ranking in Figure 6.4 helps understand the most relevant features for classes separation. Distance, braking force and delay are the most meaningful ones; surprisingly, speed and number of vehicles have less importance than expected. The left and right directions of the bars indicate the relevance in decreasing and increasing values, respectively, of the feature. The directions of distance and speed are coherent with intuition, e.g., decreasing distance increases the frequency of collision. The direction of the bar associated with the delay feature in the safety class (no collision) is however counter-intuitive as it states that safety is achieved by increasing delay. This is not uncommon in machine learning analysis as it should give unexpected insights into the problem. In this case, the delay effect is superseded by the ones of the other variables; the delay subplots in Figure 6.3 show the spread of red (collision) points over almost all the delay ranges (except very low delays). Together with Table 6.2, the ranking figures help understand how much global XAI drives a more synthetic knowledge extraction than local XAI (such as through LIME, as often used in counterfactual explanation [66]), which gives rules that are built around the point of interest and have a limited covering over the rest of the dataset. Global XAI still has local explanation property (as outlined in Table 6.3), but it may give global insight, too (as outlined later in Figure 6.6c).



(a) Feature ranking for class "non-collision". (b) Feature ranking for class "collision".

Figure 6.4: Graph of the most relevant features for the determination of the class.

6.3.3 Explanation

To determine a counterfactual explanation of $\mathcal{X}_{PL}$, 10000 points were randomly sampled from the collision class (+1) and a counterfactual was determined for each of them through **Algorithm 4**, using the Gaussian kernel-induced distance d as the distance [14]

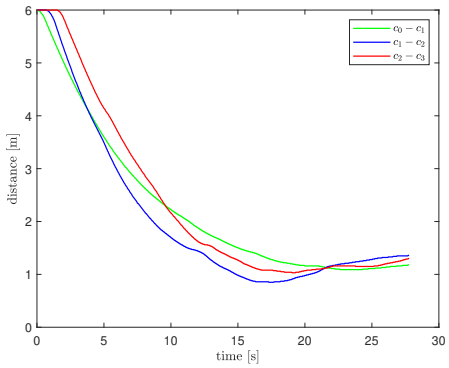
$$d(x, y) = 2 - 2k(x, y)$$

where $k(x, y) = e^{-\frac{||x-y||^2}{2\sigma^2}}$ is the Gaussian kernel. Ten examples are shown in Table 6.3.
For each row of Table 6.3, the point belonging to the collision class, classified with the SVDD and LLM and the rule, with largest covering, it satisfies; the corresponding CE, also classified with the SVDD and LLM, and the rule it satisfies is reported. The last column reports the minimum change $\Delta\mathbf{u}$ in distance and speed that allowed to move from the collision class to the non-collision class.

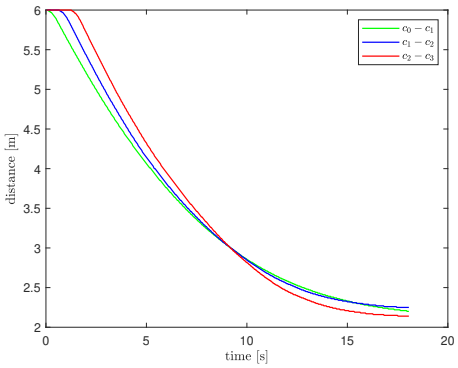
6.3.4 Validation

The validation of the counterfactuals safety is as follows: the 10000 CEs determined by **Algorithm 4** were tested by the CACC simulator [146], obtaining 7.82% error (i.e., that the determined counterfactual still brings the system into collision) and 92.18% actual counterfactuals, of which only 2.07% are found to be overestimated. Overestimation is defined with respect to a final distance larger than 10 meters[5], such a distance is found at the end of the simulation run, which is driven by the counterfactual. Figure 6.5 deals with the temporal behaviour of three significant cases; the first two (from top to bottom subplots) are optimal counterfactuals (the first with change in speed and the second one with change in distance), as they lead to a final condition which is very close to collision. The last subplot (at the bottom

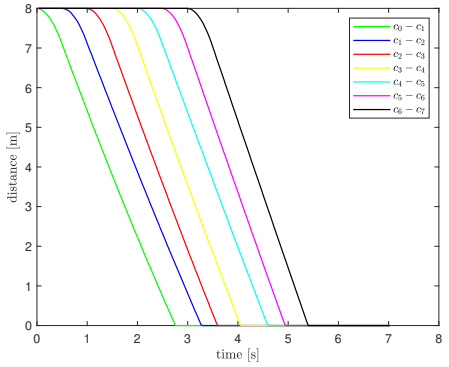
[5]A collision is considered, in the original dataset, when the distance is below the threshold of 2 meters.



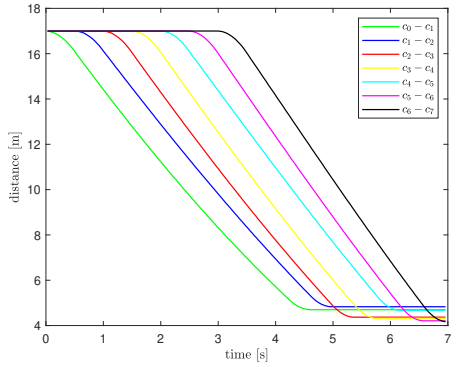
(a) Factual
Collision, $(d_0, v_0) = (6, 82)$



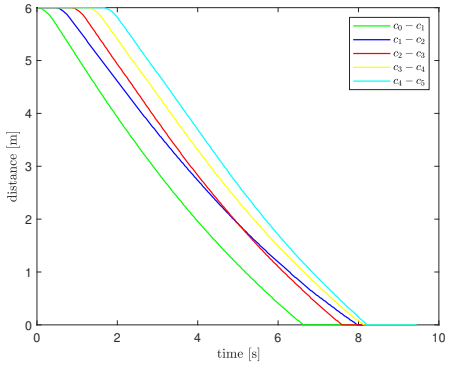
(b) Counterfactual
Non-collision, $(d_0, v0) = (6, 51)$



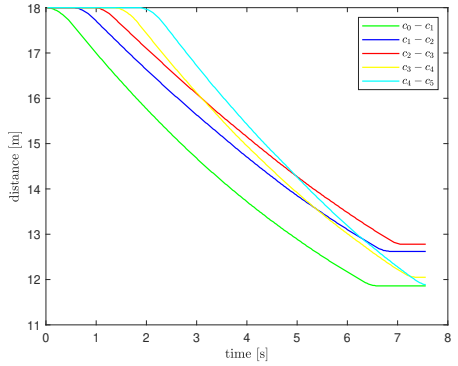
(c) Factual
Collision, $(d_0, v_0) = (8, 99)$



(d) Counterfactual
Non-collision, $(d_0, v0) = (17, 99)$

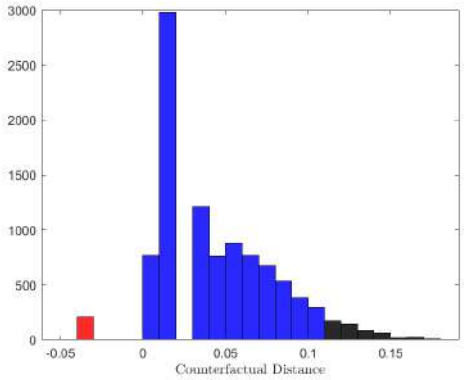


(e) Factual
collision, $(d_0, v0) = (6, 76)$

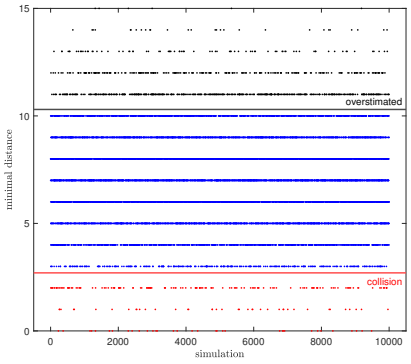


(f) Counterfactual
Non-collision, $(d_0, v0) = (18, 56)$

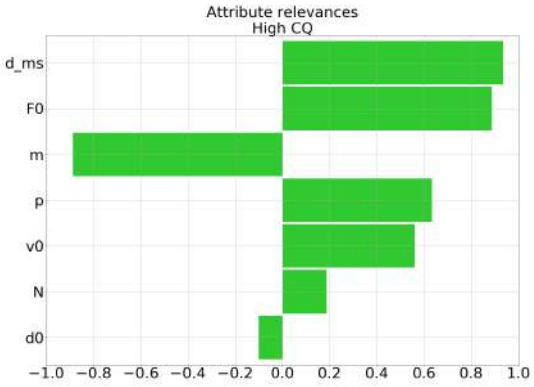
Figure 6.5: Table 6.3, row 3, 8, 9: examples of platoon distance trend of the original features and their counterfactual.



(a) CQ of extracted counterfactuals. The red bin refers to counterfactuals that are incorrect, i.e. $q < 0$. Black bins refer to counterfactuals that overestimate corrections ($q > 0.1$).



(b) Behaviour of simulations with counterfactuals extracted via **Algorithm 4**. The platoon collides when the minimum distance in the simulation is less than or equal to 2 (red dots). Black dots refer to counterfactuals that overestimate the correction (minimum distance greater than 10).



(c) Feature ranking which describes the relevance of the features in classify high value of CQ.

Figure 6.6: Metrics for validating **Algorithm 4**: (6.6a) shows the CQ of the extracted counterfactuals, (6.6b) represents the behaviour of the 10000 counterfactual simulations and (6.6c) shows the feature ranking for the class "High CQ".

of the figure) highlights an over-dimensioned counterfactual as the final distance is much larger than the boundary one (between collision and non-collision).

6.3.5 On the minimum distance

The analysis would suggest more insightful thinking on the concept of “minimum" counterfactual distance, which is ubiquitous in the literature. In the platooning application, that concept would imply "almost collision" because the counterfactual, by

construction, should lie in the safety SVDD (under the constraint of non-controllable variables), but still closest to the collision one. On the one hand, this corroborates the flexibility of counterfactual construction through the SVDD with respect to deep learning, in which the positioning of the (constrained and with minimum distance) counterfactual should be mapped into a very complex training cost. On the other hand, it would lead to other, more restricted, forms of counterfactual construction, when safety plays a crucial role. This topic is left open for future research.

6.3.6 Quality

The validation of the counterfactuals quality is as follows. The CQ of each CE is calculated, thus evidencing satisfactory statistics, as shown in Figure 6.6a, in line with simulation evidence (Figure 6.6b). The CQ metric well synthesises the overestimation issue. Recall that high QC means low quality in counterfactuals. In order to derive further knowledge extraction from the CQ analysis, the following supervised problem is defined over the CQ values and solved via the LLM. The factuals (i.e., points of the collision class, which are mapped into the corresponding counterfactuals) are mapped into two classes; the classes label CQ values under and above the 0.03 threshold. Values larger than the threshold represent overdimensioned and almost overdimensioned points, as evidenced in Figure 6.6a.
The resulting feature ranking in Figure 6.6c (for CQ>threshold) shows that high CQ samples are associated with critical factuals, namely, with increasing delay, leader acceleration (force divided by the mass), loss, speed and number of vehicles as well as decreasing distance. The rationale of the conditions relies on the fact that critical factuals need to go deeper inside the destination class (thus leading to larger CQ) to replace the original conditions of collision into new safety ones. Moreover, the rules identifying high CQ may drive further optimisation of the respective counterfactuals, e.g., through a finer granularity of the grid in a reduced search space, identified by the ruleset itself [147]. This is left open for future research as well.

6.3.7 Discussion

This study aims to define a new method for generating local explanations by defining counterfactuals from observations characterized by controllable and non-controllable features. Nemirovsky et al. [121] first introduced the concept of CEs with controllable and non-controllable features in a diabetes prediction algorithm, however they first applied counterfactual search to all the features and then they removed the perturbations related to non-controllable features like age and the number of pregnancies. In this study, controllable and non-controllable features are handled in a more straightforward way, since the search for counterfactuals is instead done by perturbing only the controllable features (i.e., d_0 and v_0) in the kernel space, keeping the non-controllable variables fixed. Most of the recently proposed methods are deep learning based [121, 82], thus requiring more complex architectures and higher computational cost for training. The use of TC-SVDD allows to define the two regions with a reduced computational cost, yet still achieving more than satisfactory

accuracy (e.g., $> 85\%$). Furthermore, the additional rule-based description of the SVDD regions provides transparency to the point classification process, allowing for a robust validation of correctness and consistency of the generated CEs. Specifically, as shown in Table 6.3, in the platooning example, CEs are generally associated with greater initial distance and reduced initial velocity of the platoon. Moreover, the quality of explanations have been evaluated in terms of distance from the region associated with the opposite outcome. The optimal CE of $\boldsymbol{x}$ is the point, with opposite class, located at minimum distance from $\boldsymbol{x}$. The introduction of a quality metric (CQ) allows to verify the correctness of CEs, generated with the proposed numerical approximation, since a distance greater than zero ensures the non-intersection between the two SVDD regions, thus the belonging of the CE to the correct class, with a certain level of confidence defined by the TC-SVDD (i.e., 88% in the platooning example) and a distance close to zero ensures the minimum distance requirement. Figure 6.6b shows CQ values for the generated platooning CEs, demonstrating the effectiveness of the proposed method, as most of the points are associated to a low but positive CQ value. Indeed, almost 40% of the points are associated to CQ lower than 0.02 and about 92% of the points present CQ lower than 0.1. Unlike previous works in this area, the validation of the generated counterfactuals is not only based on class prediction via SVDD, but further supported by validation via simulations. In fact, the attribution of the point to the correct class according to the prediction of the previously trained model does not guarantee its real belonging to that class, because of the existence of a certain number of false positives and false negatives that, even if minimized, should not be neglected. The validation process through the CACC simulator (see 6.6a) has proven that the generated CEs are descriptive of the non-collision class with a more than satisfactory accuracy, and that only a small part of the generated points overestimates the minimum distance. Hence, the use of CE in truck platooning results applicable to the generation of control algorithms, based on the correction of the system dynamics, to prevent collisions. Nevertheless, the method results applicable to a wide range of applications, not only to cyberphysical systems (e.g, disease prediction and prevention, fraud detection...). For example, CEs generated through the application of variable distance perturbations could be useful to provide an estimate of risk in the case of chronic diseases, such as diabetes, and contribute to the formulation of preventive strategies. In fact, CEs generated at minimum distance are associated to an higher risk of developing the disease, whereas CEs generated at a progressively increasing distance are associated with a lower risk. The proposed framework, proves to be trustworthy, thanks to the use of the LLM, which allows to characterize the extracted CEs through readily interpretable rules that can be easily understood and validated by application domain experts, even if they have no prior knowledge in the field of artificial intelligence.

Chapter 7

Multi-Counterfactual eXplanations via SVDD

This Chapter is dedicated to the formulation of the extended version to the multi-class case of SVDD and how to define multi-class counterfactual explanations. Beyond the developmnet of the new theory, examples to show the efficacy of the methodology are proposed.

The training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is composed by m classes of objects of different sizes $n_1, n_2, \ldots, n_m$ $(n_1 + n_2 + \ldots + n_m = n)$, labelled according to their class

$$\mathbf{y} = \left[\begin{array}{cccccccccc} 1 & \ldots & 1 & 2 & \ldots & 2 & \ldots & m & \ldots & m \end{array} \right]^\top.$$

In order to find the m hyperspheres with minimum total volume, we should minimize the total volume of the m hyperspheres with the constraint that, for each object, (i) the distance between the center of one hypersphere and the object is smaller than the radius of that hypersphere (i.e., the object belongs to a specific output class) and (ii) the object should not fall into other hyperspheres (i.e., the object should not belong to other output classes).
Let $\mathbf{a}_k$ and R_k denote the center and radius of the hypersphere k. To allow a flexible description of the hyperspheres we introduce $\varphi : \mathcal{X} \longrightarrow \mathcal{V}$, a *feature map* from the space of the input features $\boldsymbol{x} \in \mathcal{X}$ to an higher dimensional inner product space $\mathcal{V}$.
Searching for hyperspheres of minimum volume that satisfy the above constraints means finding the solution of the following optimization problem

$$\min\ F(R_k; \mathbf{a}_k) = \sum_{k=1}^{m} R_k^2 \tag{7.1a}$$

$$\text{s.t.} \quad \left\| \varphi(\boldsymbol{x}_i^k) - \mathbf{a}_k \right\|^2 \leq R_k^2,\ i \in [n_k], \forall k \tag{7.1b}$$

$$\left\| \varphi(\boldsymbol{x}_i^k) - \mathbf{a}_h \right\|^2 \geq R_h^2,\ i \in [n_k], \forall h \neq k \tag{7.1c}$$

We can follow the classical approach as in [11], which consists in reducing (7.1) to a quadratic programming problem. To allow for the possibility of outliers in the training set, the distance from an object belonging to class k, $\varphi(\boldsymbol{x}_i^k)$, to its own centre $\mathbf{a}_k$ should not be strictly smaller than R_k^2 , but larger distances should be penalized, and the distance from $\varphi(\boldsymbol{x}_i^k)$ to the other centres $\mathbf{a}_h$, $h{\neq}k$, should not be strictly larger than R_h^2, i.e. smaller distances should be penalized. Therefore we introduce slack variables $\boldsymbol{\xi}^{kk} \geq 0, \boldsymbol{\xi}^{kh} \geq 0$ and the minimization problem changes into

$$\min\ F(R_k; \mathbf{a}_k; \xi^{kh}) = \sum_{k=1}^{m} R_k^2 + \sum_{k=1}^{m}\sum_{h=1}^{m} C_{kh} \sum_{i=1}^{n_k} \xi_i^{kh} \tag{7.2a}$$

$$\text{s.t.} \quad \left\|\varphi(\boldsymbol{x}_i^k) - \mathbf{a}_k\right\|^2 \leq R_k^2 + \xi_i^{kk},\ i \in [n_k], \forall k \tag{7.2b}$$

$$\left\|\varphi(\boldsymbol{x}_i^k) - \mathbf{a}_h\right\|^2 \geq R_h^2 - \xi_i^{kh},\ i \in [n_k], \forall h \neq k \tag{7.2c}$$

$$\text{and } \xi_i^{kk} \geq 0\ \ \forall k, \xi_i^{hk} \geq 0\ \ \forall h \neq k \tag{7.2d}$$

where the parameter C_{kh} controls the misclassification error between the classes. Now, we consider the dual problem of (7.2) by incorporating the constraints (7.2b) and (7.2c) into (7.2a) with the introduction of Lagrange multipliers

$$\begin{aligned} & L(R_k; \mathbf{a}_k; \boldsymbol{\xi}^{kk}, \boldsymbol{\xi}^{kh}; \boldsymbol{\alpha}^{kk}, \boldsymbol{\alpha}^{kh}; \boldsymbol{\gamma}^{kk}, \boldsymbol{\gamma}^{kh}) \\ = & \sum_{k=1}^{m} R_k^2 + \sum_{k=1}^{m}\sum_{h=1}^{m} C_{kh} \sum_{i=1}^{n_k} \xi_i^{kh} \\ - & \sum_{k=1}^{m}\sum_{i=1}^{n_k} \alpha_i^{kk} \left(R_k^2 + \xi_i^{kk} - \left\|\varphi(\boldsymbol{x}_i^k) - \mathbf{a}_k\right\|^2\right) \\ - & \sum_{h\neq k}\sum_{i=1}^{n_h} \alpha_i^{kh} \left(\left\|\varphi(\boldsymbol{x}_i^k) - \mathbf{a}_h\right\|^2 - R_h^2 + \xi_i^{kh}\right) \\ - & \sum_{k=1}^{m}\sum_{i=1}^{n_k} \gamma_i^{kk}\xi_i^{kk} - \sum_{h\neq k}\sum_{i=1}^{n_h} \gamma_i^{kh}\xi_i^{kh} \end{aligned} \tag{7.3}$$

with the Lagrange multipliers $\boldsymbol{\alpha}^{kk}, \boldsymbol{\alpha}^{kh}, \boldsymbol{\gamma}^{kk}, \boldsymbol{\gamma}^{kh} \geq 0$ (7.4). In the dual form, L should be maximized with respect to the Lagrange multipliers so setting partial derivatives to zero gives the new constraints

$$\frac{\partial L}{\partial R_k} = 0 \Rightarrow \sum_{i=1}^{n_k} \alpha_i^{kk} - \sum_{h\neq k}\sum_{i=1}^{n_h} \alpha_i^{kh} = 1 \tag{7.5}$$

$$\frac{\partial L}{\partial \mathbf{a}_k} = 0 \Rightarrow \mathbf{a}_k = \sum_{i=1}^{n_k} \alpha_i^{kk}\varphi(\boldsymbol{x}_i^k) - \sum_{h\neq k}\sum_{i=1}^{n_h} \alpha_i^{kh}\varphi(\boldsymbol{x}_i^h) \tag{7.6}$$

$\forall k \in [m]$ and $\forall h \neq k$. And with respect to the slack variables

$$\frac{\partial L}{\partial \xi_i^{ss}} = 0 \Rightarrow C_{ss} - \alpha_i^{ss} - \gamma_i^{ss} = 0 \Rightarrow 0 \leq \alpha_i^{ss} \leq C_{ss} \tag{7.7}$$

$$\frac{\partial L}{\partial \xi_i^{st}} = 0 \Rightarrow C_{st} - \alpha_i^{st} - \gamma_i^{st} = 0 \Rightarrow 0 \leq \alpha_i^{st} \leq C_{st} \tag{7.8}$$

$\forall s \in [m]$ and $\forall t \neq s$ respectively.
Substituting (7.5) and (7.6) in (7.4) the Lagrangian in the dual takes this form

$$\begin{aligned} L &= \sum_{k=1}^{m}\sum_{i=1}^{n_k} \alpha_i^{kk}\big(\varphi(\boldsymbol{x}_i^k)\cdot\varphi(\boldsymbol{x}_i^k)\big) \\ &- \sum_{h\neq k}\sum_{i=1}^{n_k} \alpha_i^{kh}\big(\varphi(\boldsymbol{x}_i^k)\cdot\varphi(\boldsymbol{x}_i^k)\big) \\ &- \sum_{i=1}^{m}\sum_{i,j=1}^{n_k} \alpha_i^{kk}\alpha_j^{kk}\big(\varphi(\boldsymbol{x}_i^k)\cdot\varphi(\boldsymbol{x}_j^k)\big) \\ &- \sum_{h\neq k}\sum_{i,j=1}^{n_k} \alpha_i^{kh}\alpha_j^{kh}\big(\varphi(\boldsymbol{x}_i^k)\cdot\varphi(\boldsymbol{x}_j^k)\big) \\ &+ 2\sum_{h\neq k}\sum_{i=1}^{n_k}\sum_{j=1}^{n_h} \alpha_i^{kk}\alpha_j^{kh}\big(\varphi(\boldsymbol{x}_i^k)\cdot\varphi(\boldsymbol{x})_j^h\big) \end{aligned} \tag{7.9}$$

The maximization of (7.10) under the constraints (7.4)-(7.5) and (7.7)-(7.8) gives the set of $\boldsymbol{\alpha}^{kk}, \boldsymbol{\alpha}^{kh}$ $\forall k \in [m]$, $\forall h \neq k$ ($\boldsymbol{\gamma}^{kk}$ and $\boldsymbol{\gamma}^{kh}$ can be eliminated by exploiting their positivity and the first-order conditions on the slack variables).
Depending on the position of the training objects in the feature space, the Lagrange multipliers take on different values in the way the training objects do or do not satisfy the constraints (7.2b) and (7.2c)

$$\begin{aligned} \left\|\varphi(\boldsymbol{x}_i^k) - \mathbf{a}_k\right\|^2 < R_k^2 &\Rightarrow \alpha_i^{kk} = 0 \\ \left\|\varphi(\boldsymbol{x}_i^k) - \mathbf{a}_h\right\|^2 > R_h^2 &\Rightarrow \alpha_i^{kh} = 0 \\ \left\|\varphi(\boldsymbol{x}_i^k) - \mathbf{a}_k\right\|^2 = R_k^2 &\Rightarrow 0 < \alpha_i^{kk} < C_{kk} \\ \left\|\varphi(\boldsymbol{x}_i^k) - \mathbf{a}_h\right\|^2 = R_h^2 &\Rightarrow 0 < \alpha_i^{kh} < C_{kh} \\ \left\|\varphi(\boldsymbol{x}_i^k) - \mathbf{a}_k\right\|^2 > R_k^2 &\Rightarrow \alpha_i^{kk} = C_{kk} \\ \left\|\varphi(\boldsymbol{x}_i^k) - \mathbf{a}_h\right\|^2 < R_h^2 &\Rightarrow \alpha_i^{kh} = C_{kh} \end{aligned} \tag{7.10}$$

$\forall k \in [m]$ and $\forall h \neq k$ respectively.
Then, according with the literature around SVDD [11], the objects $\boldsymbol{x}_i^k$ with $\alpha_i^{kk} > 0$

and $\alpha_i^{kh} > 0$ are called *Support Vectors* (SVs) for the Class k.
By definition, (7.11), the radius R_k is the distance from the center $\mathbf{a}_k$ of the hypersphere to any of the SVs of Class k with Lagrange multipliers strictly minor than the parameters $C_{k\{\cdot\}}$. Therefore

$$\begin{aligned} R_k^2 &= \left\|\varphi(\boldsymbol{x}_s^k) - \mathbf{a}_k\right\|^2 = \big(\varphi(\boldsymbol{x}_s^k)\cdot\varphi(\boldsymbol{x}_s^k)\big) \\ &- 2\sum_{i=1}^{n_k}\alpha_i^{kk}\big(\varphi(\boldsymbol{x}_s^k)\cdot\varphi(\boldsymbol{x}_i^k)\big) \\ &+ 2\sum_{h\neq k}\sum_{i=1}^{n_h}\alpha_i^{kh}\big(\varphi(\boldsymbol{x}_s^k)\cdot\varphi(\boldsymbol{x}_i^h)\big) \\ &+ \sum_{i,j=1}^{n_k}\alpha_i^{kk}\alpha_j^{kk}\big(\varphi(\boldsymbol{x}_i^k)\cdot\varphi(\boldsymbol{x}_j^k)\big) \\ &- 2\sum_{h\neq k}\sum_{i=1}^{n_k}\sum_{j=1}^{n_h}\alpha_i^{kk}\alpha_j^{kh}\big(\varphi(\boldsymbol{x}_i^k)\cdot\varphi(\boldsymbol{x}_j^h)\big) \\ &+ \sum_{h\neq k}\sum_{i,j}^{n_h}\alpha_i^{kh}\alpha_j^{kh}\big(\varphi(\boldsymbol{x}_i^h)\cdot\varphi(\boldsymbol{x}_j^h)\big) \end{aligned} \tag{7.11}$$

for any SVs $\varphi(\boldsymbol{x}_s^k)$ of Class k with $0 < \alpha_i^{kk} < C_{kk}$ or $0 < \alpha_i^{kh} < C_{kh}$, for $h \neq k$.
To test an object $\mathbf{t}$ it is necessary to calculate its distance from the centre of the hypersphere k, i.e.

$$\begin{aligned} d_k &\doteq \|\mathbf{t} - \mathbf{a}_k\|^2 \\ &= \big(\varphi(\mathbf{t})\cdot\varphi(\mathbf{t})\big) - 2\sum_{i=1}^{n_k}\alpha_i^{kk}\big(\varphi(\mathbf{t})\cdot\varphi(\boldsymbol{x}_i^k)\big) \\ &+ 2\sum_{h\neq k}\sum_{i=1}^{n_h}\alpha_i^{kh}\big(\varphi(\mathbf{t})\cdot\varphi(\boldsymbol{x}_i^h)\big) \\ &+ \sum_{i,j=1}^{n_k}\alpha_i^{kk}\alpha_j^{kk}\big(\varphi(\boldsymbol{x}_i^k)\cdot\varphi(\boldsymbol{x}_j^k)\big) \\ &- 2\sum_{h\neq k}\sum_{i=1}^{n_k}\sum_{j=1}^{n_h}\alpha_i^{kk}\alpha_j^{kh}\big(\varphi(\boldsymbol{x}_i^k)\cdot\varphi(\boldsymbol{x}_j^h)\big) \\ &+ \sum_{h\neq k}\sum_{i,j}^{n_h}\alpha_i^{kh}\alpha_j^{kh}\big(\varphi(\boldsymbol{x})_i^h\cdot\varphi(\boldsymbol{x}_j^h)\big) \end{aligned} \tag{7.12}$$

a test object $\mathbf{t}$ is accepted by the following criterion:

1. If $d_k \leq R_k^2$ and $d_k > R_h^2\ \forall h \neq k$, then $\mathbf{t}$ belongs to Class k;

2. If $d_k \leq R_k^2$ and $d_h < d_k \ \forall h \neq k$, then $\mathbf{t}$ belongs to Class h;

3. If $d_k > R_h^2 \ \forall h$, then $\mathbf{t}$ is unclassified.

That is, the distances between all samples in each class and the center should be smaller than the radius of the corresponding hypersphere and the distances between all samples in each class and the centers of other classes should be larger than the radius of the corresponding hypersphere. And if a new sample belongs to more than a hypersphere, the sample is assigned to the class corresponding to the minimum distance. In any other case the sample is unclassified. Figure 7.1 clearly shows the behavior of the algorithm for linearly separated data: each sphere encloses the points related to an output class by minimizing its volume and excluding unclassified points. It is worth to underline the following remarks:

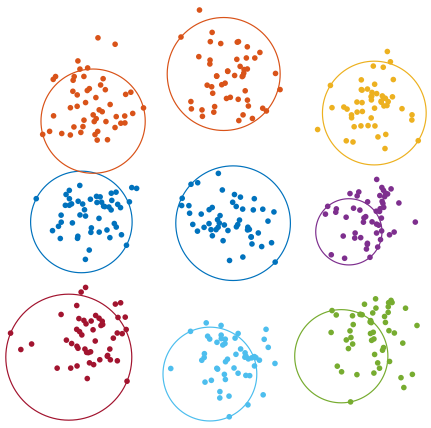


Figure 7.1: MC-SVDD applied to 9 classes extracted randomly from Gaussian distributions with different means and variances. Here, a linear MC-SVDD has been trained by fixing 9^2 parameters C_{kh} to control the trade-off between class covering and error between the classes.

Remark 7.1. In order to obtain a more compact form of the Lagrangian L, and to make it clear that the problem is quadratic, we define these quantities for all $k \in [m]$

$$\boldsymbol{\alpha}^k \doteq \left[\ \boldsymbol{\alpha}^{k1}, \boldsymbol{\alpha}^{k2}, \ldots, \boldsymbol{\alpha}^{km} \ \right]^\top \ , \quad \boldsymbol{\alpha} \doteq \left[\ \boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^m \ \right]^\top$$

$$\mathbf{y}^k \ = \ \left[\ y_1^k \quad y_2^k \quad \ldots \quad y_n^k \ \right]^\top ,$$

$$\text{where } y_i^k = \begin{cases} +1 & \text{if } y_i = k \\ -1 & \text{if } y_i \neq k \end{cases} \quad \forall i \in [n].$$

Defined then, for all $k \in [m]$

$$\Phi_k \doteq \left[\varphi(\boldsymbol{x}_1^k) \quad \varphi(\boldsymbol{x}_2^k) \quad \dots \quad \varphi(\boldsymbol{x}_n^k) \right], \tag{7.13}$$
$$D_k \doteq \text{diag}\{y_1^k, y_2^k, \dots, y_n^k\}, \tag{7.14}$$
$$K_k \doteq \Phi_k^\top \Phi_k, \tag{7.15}$$

and $K_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^\top \varphi(\boldsymbol{x}_j),\ i \in [n], j \in [n]$, is the kernel matrix which satisfies the Mercer's theorem [1].
Then let them be

$$H_k \doteq 2D_k K_k D_k,$$
$$f_k \doteq D_k \text{diag}(K_k).$$

Finally, defining

$$H \doteq \begin{pmatrix} H_1 & & & \\ & H_2 & & \\ & & \ddots & \\ & & & H_m \end{pmatrix},\ f \doteq \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_m \end{bmatrix}$$

we obtain that the Lagrangian L, (7.10), can be rewritten as

$$L = -\frac{1}{2}\boldsymbol{\alpha}^\top H \boldsymbol{\alpha} + f^\top \boldsymbol{\alpha}, \tag{7.16}$$

i.e. L is a quadratic form, that can be easily maximized with a quadratic optimizer.

7.1 MUCH: MUlti Counterfactual via Halton sampling

A dataset $\mathcal{D}$ can be described by a subset of modifiable features $\mathbf{u}$ and a subset of non-modifiable features $\mathbf{z}$. As a consequence, an observation $\boldsymbol{x} \in \mathcal{D}$ can be defined as

$$\boldsymbol{x} = \left(u^1, u^2, \dots, u^p, z^1, z^2, \dots, z^q\right) \in \mathbb{R}^{p+q=N}$$

MC-SVDD is applied to obtain m classification regions defined as follows:

$$S_i \doteq \{\boldsymbol{x} \in \mathbb{R}^N : \|\boldsymbol{x} - \mathbf{a}_i\|^2 \leq R_i^2, \|\boldsymbol{x} - \mathbf{a}_j\|^2 \geq R_j^2; j \in [m]; j \neq i\} \tag{7.17}$$

where $R_i^2, R_j^2, \mathbf{a}_i, \mathbf{a}_j$ represent the radii and the centers of the spheres, as defined in Section 6.1. Once the m classification regions are defined, the search for a counterfactual explanation of an observation $\boldsymbol{x}_f \in S_i$, called *factual*, consists of determining the minimum joint variation $\Delta\mathbf{u}^*$ of the modifiable variables to obtain the closest observation

$$\boldsymbol{x}_{f,j}^* \doteq (\mathbf{u} + \Delta\mathbf{u}^*, \mathbf{z})_{f,j} \tag{7.18}$$

that belongs to class S_j different from the original class S_i.
Specifically, $\Delta\mathbf{u}^*$ is estimated by solving the following minimization problem: For all $j \in [m], j \neq i$

$$\min_{\Delta\mathbf{u}\in\mathbb{R}^p} \quad d\big(\boldsymbol{x}_f, (\mathbf{u}+\Delta\mathbf{u}, \mathbf{z})_{f,j}\big) \tag{7.19a}$$

$$\text{subject to} \quad \|(\mathbf{u}+\Delta\mathbf{u}, \mathbf{z})_{f,j} - \mathbf{a}_j\|^2 \leq R_j^2 \tag{7.19b}$$

$$\|(\mathbf{u}+\Delta\mathbf{u}, \mathbf{z})_{f,j} - \mathbf{a}_k\|^2 \geq R_k^2, \quad \text{with } k \in [m] \text{ and } k \neq j \tag{7.19c}$$

where d is the selected distance metrics (e.g., the Euclidean norm), (7.19b) constraints $\boldsymbol{x}^*$ to lie inside S_i and (7.19c) constraints $\boldsymbol{x}^*$ to lie outside all the regions $S_k \neq S_j$. It is worth noting that, for each factual $\boldsymbol{x}_f \in S_i$, we can find a set $\mathbf{C}_f = \{\mathbf{x}^*_{f,j} \mid j \in [m]; j \neq i\}$ of $m-1$ counterfactual explanations, that is, one for each class j different from i. In other words, for a set of factuals $\mathbf{F}_i$ we obtain a set of counterfactual explanations $\mathbf{E}$ with maximum size $(|\mathbf{F}_i|, m-1)$.

7.1.1 Numerical solution

Since each S_j theoretically includes an infinite set of real points, a numerical approximation is necessarily introduced whereby counterfactual explanations are sought in a sampled region obtained by applying quasi-random Halton sampling [43][1]. Since counterfactual explanations are searched among a finite set of points, the availability and minimality of each explanation depends on the density of the sampling. However, the higher the number of points in the sampled region, the higher the computational cost. As a consequence, a trade-off between accuracy and runtime must be reached. Counterfactual explanations are extracted for each factual observation belonging to each class. Once a factual $\boldsymbol{x}_f \in \mathbf{F}_i$, $i \in [m]$ is defined, the algorithm returns the set of counterfactuals $\mathbf{C}_f$, i.e., each counterfactual $\boldsymbol{x}^*_{f,j}, j \in [m] \setminus \{i\}$.

Algorithm 5 `MUCH`
Dataset $\mathcal{D}$ is divided in training set $\mathcal{D}_{tr}$ and validation set $\mathcal{D}_{vl}$.
A MC-SVDD is performed on $\mathcal{D}_{tr}$ and validated on $\mathcal{D}_{vl}$, getting $S_1, S_2, \ldots, S_m$. A set of factuals related to the class i, $\mathbf{F}_i$, is chosen.

1 $\quad \mathbf{C_{F_i}} = [\,]$
2 $\quad$ **for** $\mathbf{x}_f = (\mathbf{u}_f, \mathbf{z}_f) \in \mathbf{F_i}$
2.1 $\quad\quad \mathbf{C}_f = [\,]$

[1]Halton is a low discrepancy sequence generator; other generators of this type, such as Sobol, may be applicable in the sampling step of the algorithm.

2.2 **for** $j \in [m]$, $j \neq i$
2.2.1 **Sample** quasi-randomly $\tilde{S}_j$
2.2.2 $d_f = d\big(\boldsymbol{x}_f, \tilde{S}_{j_{|\mathbf{z}=\mathbf{z_f}}}\big)$
2.2.3 $\boldsymbol{x}'_f = \min(d_f)$
2.2.4 **if**$\big(\boldsymbol{x}_f \in S_i \;\&\; \boldsymbol{x}'_f \in S_j\big)$
2.2.4.1 $\mathbf{C}_f = \mathbf{C}_f \cup \{\mathbf{x}'_f\}$
2.2.5 **end**
2.2.6 $\mathbf{C}_{\mathbf{F_i}} = \mathbf{C}_{\mathbf{F_i}} \cup \mathbf{C}_f$
2.3 **end**
2.4 **end**
3 **return** $\mathbf{C}_{\mathbf{F}_i}$

The first step of the MUCH algorithm[2] (MUltiCounterfactual via Halton sampling) (**Algorithm 5**) is the classification of data by MC-SVDD, which defines m regions S_i, $i \in [m]$, into which data are classified. The MC-SVDD algorithm is trained on $\mathcal{D}_{tr}$ and validated on $\mathcal{D}_{vl}$, each belonging to the same probability distribution of the data, recovering the best classification after hyperparameter tuning. Then, for each region S_i a randomly sampled region $\tilde{S}_i$ is constructed: this region is the one designated to the numerical search for counterfactuals of class $j \neq i$, i.e., for each factual $\boldsymbol{x}_f$ the respective counterfactual related to the class $j \neq i$, $\boldsymbol{x}^*_{f,j}$, is searched in $\tilde{S}_j$. Among all points in the sampled region $\tilde{S}_j$, the one that minimizes the distance d w.r.t factual $\boldsymbol{x}_f$ is chosen. The distance d plays a key role in the search for counterfactuals as changing the distance may changes the returned counterfactuals. The most natural choice of distance is the distance induced by the classification kernel:

$$d(\boldsymbol{x}, \mathbf{y}) = k(\boldsymbol{x}, \boldsymbol{x}) - 2k(\boldsymbol{x}, \mathbf{y}) + k(\mathbf{y}, \mathbf{y}).$$

The reason for this choice is motivated by the fact that the topology defined by the kernel in the classification affects the relationship between the points in the sampled regions, so keeping the same distance relationship would help the algorithm find the best counterfactual explanation. The estimation of the computational cost of MUCH can be easily retrieved from the computational cost of the binary counterfactual generator algorithm proposed in [159]. Denoting with n the number of points, with d the number of features and m the number of classes, the computational cost of MC-SVDD, that is, O(MC-SVDD) is estimated in $O\left(m\left(\max(n,d)\min(n,d)^2\right)\right)$. In accordance with [159], the computational cost related to the counterfactuals search, for each set of factuals $\mathbf{F}_i$, is $O\left(\max\left(\sum_{j\neq i} q_j, |\mathbf{F}_i| \max\left(D, \sum_{j\neq i} \tilde{s}_j\right)\right)\right)$, where $O(q_j)$ is the computational cost of the random sampling of $\tilde{S}_j$ [27], $O(D)$ is the computational cost for the computation of the distance d [16] and $O(\tilde{s}_j)$ is the computational cost of the research of the minimum of the vector of distances relative to the $j-$th

[2] `https://github.com/AlbiCarle/MUCH.git`

random sampling ($\tilde{S}_j$) [4]. So, considering all m classes, the computational cost of the counterfactuals search, O (SCF), can be estimated in

$$O\left(m\left(\max\left(\sum_{j\neq i} q_j, |\mathbf{F}_i| \max\left(D, \sum_{j\neq i} \tilde{s}_j\right)\right)\right)\right).$$

Finally, the total computational cost of MUCH can be estimated with O (MUCH) = O (max(MC-SVDD, SCF)). The complete procedure for generation of a set of explanations is summarized in Fig. 7.2.

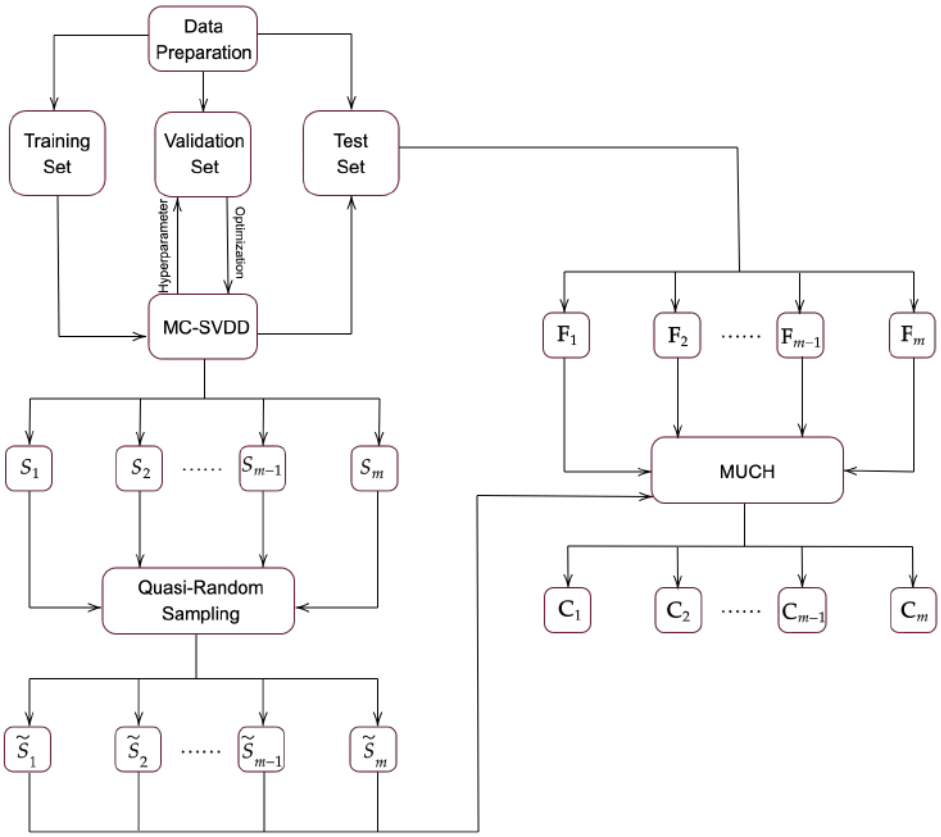


Figure 7.2: Diagram of the counterfactual extraction procedure.

7.1.2 Counterfactual quality

As reported in a recent review by Guidotti [161], counterfactual explanations should fulfill a set of ideal properties and adherence to these properties shall be assessed, for a set of factuals, in terms of appropriate evaluation metrics such as availability,

actionability, similarity and discriminative power. *Availability* measures the number of counterfactuals actually returned by the counterfactual explainer for each class and it can be measured as the ratio between the number of counterfactuals of class j i.e., $|\mathbf{E}_j|$ and the total number of factuals of class i, i.e., $|\mathbf{F}_i|$. *Actionability* measures the ability of counterfactual explanations to vary only modifiable features and it is calculated, for each class j, as the ratio of the number of constrained features and the total number of non-modifiable features i.e., $|\mathbf{z}|$. *Similarity* evaluates the average distance (e.g., Euclidean) between each factual in $\mathbf{F}_i$ and the corresponding counterfactual explanations in $\mathbf{E_j}$. In order to be similar, the distance between these two points should be lower than a fixed threshold ε. To evaluate similarity, data points were normalized between 0 and 1 and the computed distance was compared to the maximum theoretical distance in the standardized modifiable-feature space (i.e., $\sqrt{|\mathbf{u}|}$) and represented in terms of average and 95% confidence interval (C.I.). Finally, *discriminative power* measures the ability to distinguish points of the factual class in S_i from counterfactuals in $\mathbf{E}_j$. This metrics was estimated in this study by evaluating the accuracy of a k-Nearest Neighbor (KNN) classifier trained on a dataset including the counterfactuals in $\mathbf{E}_j$ and real data points in S_i. Discriminative power was then computed as the average test accuracy obtained with 5-Fold cross-validation. In a multi-class classification problem, such as the one considered in the next section, where $|\mathbf{C}_f| > 1$, each evaluation metric can be considered as the average value obtained across the $m - 1$ set of counterfactuals.

7.2 Clarifying example: the FIFA dataset

7.2.1 Dataset description

FIFA is one of the most famous football videogames in the world. The FIFA dataset[3] includes latest edition FIFA attributes related to more than 17000 players from different football leagues. In this study, a subset of 50 attributes were selected from the initial set of 89 attributes. Specifically, the attributes related to the player's physical and athletic characteristics were retained, whereas those not relevant (e.g., team, graphical visualization) were discarded. Besides age, height and weight, the selected attributes can be summarized in three main categories: mental, physical and technical skills. These attributes depict different aspects of the player's individual abilities and they are usually represented in terms of rating, on a scale from 1 to 100. Attributes can be grouped in three categories, according to the ability to which they relate: Mental, Physical and Technical Skills. Moreover, the main attributes can be combined in 6 fundamental attributes, namely *Pace* (55% sprint speed, 45% acceleration), *Shooting* (ability to score: 45% finishing, 20% shot power, 20% long shots, 5% penalties, 5% positioning, 5% volleys), *Passing* (capability to successfully pass the ball to other teammates:35% short passing, 20% vision, 20% crossing, 15% long passing, 5% curve, 5% free kick accuracy), *Dribbling* (50% drib-

[3]Retrieved [November 2022] from `https://www.kaggle.com/datasets/cashncarry/fifa-23-complete-player-dataset`

bling, 35% ball control, 10% agility, 5% balance), *Defending* (ability to intercept the ball and mark the opponent: 30% marking, 30% sliding tackle, 20% interception, 10% heading accuracy, 10% sliding tackle) and *Physical* (50% strength, 25% stamina, 20% aggression, 5% jumping). These key attributes can be directly derived from the others, and for this reason, only the 44 secondary attributes were considered as input features. Given these input attributes, the classification task consisted in predicting the correct player's position among 4 possible classes: *Midfielder* (MF), *Defender* (DE), *Forward* (FO), or *Goalkeeper* (GK). To obtain a balanced dataset, 2000 records were extracted for each player's position (8000 records in total). The dataset was then splitted in training set (70%, 5600 records) and test set (30%, 2400 records). The parameters of MC-SVDD were optimized by performing a cross validation on the training set, as explained in Section 6.1. MC-SVDD with the best combination of hyperparameters was then tested on the remaining data. Table 7.1 shows the MC-SVDD training and test classification performance.

Table 7.1: Classification performance: FIFA dataset

	%OUT	ACC	F1-SCORE	Cohen's Kappa
Training	0.59%	78.03%	73.08%	0.71
Test	1.25%	77.50%	72.99%	0.70

Specifically, the performance was evaluated in terms of classification accuracy, macro-averaged F1-score (i.e., the mean of F1-scores computed by class), Cohens Kappa Coefficient [3] (i.e., the level of agreement between ground truth and predicted values) and the percentage of unclassified points (i.e., points lying outside all m SVDD regions). Accuracy and F1-SCORE are satisfactory as they are both are above 72%, moreover there is no presence of overfitting as these values remain stable even when the model is applied to test data. The percentage of unclassified points is really small, meaning that the spherical regions identified by MC-SVDD are able to enclose almost all points and the presence of anomalous points in the selected dataset is limited.

As it can be noticed from Fig. 7.3, classes DE, FO, and GK can be accurately classified. On the contrary, class MF is more difficult to discriminate. Indeed, the single class F1-score on the test set is more than acceptable when considering DE, FO and GK (i.e., 84.78%, 79.24%, and 100%, respectively), whereas it is noticeably lower when considering MF (27.96%). This is due to the fact that points in the MF class are easily confused with those in DE and FO classes as the characteristics of MF players are, in practice, intermediate between those of DE and FO players. It can also be observed that GK are perfectly distinguishable from footballers in other game positions, because of the peculiar skills that this kind of player must demonstrate.

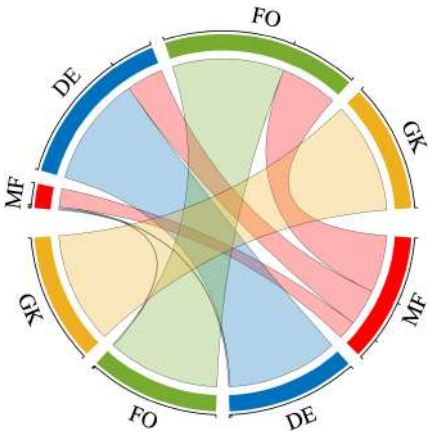


Figure 7.3: Chord diagram representation of the confusion matrix corresponding to the classification of the FIFA testing dataset.

7.2.2 Multi-counterfactuals generation

7.2.2.1 Setting

To evaluate the MUCH approach, a set of counterfactuals is generated starting from a set of points belonging to the test set. Specifically, given a player belonging to the chosen factual class and the corresponding set of attributes, the algorithm aims to find a counterfactual in each of the other classes, that is, to find the minimal changes in the player's attributes able to change his preferable position. Once $\mathbf{F}_i$ has been defined, a sufficiently large set of candidate counterfactuals is obtained by sampling 10000 points for each of the $m-1$ MC-SVDD regions using Halton sampling (see Section 7.1.1). As already mentioned, $\mathbf{F}_i$ is a set of test data points, but the corresponding counterfactuals explanations does not necessarily belong to the original dataset. Indeed, counterfactuals explanations as returned by the proposed algorithm are plausible combinations of features sampled inside the classification regions. Thus, the proposed approach is categorized as *exogenous* [161].
Age and *height* were considered as non-modifiable features, hence they were constrained during counterfactual search. Actually, counterfactuals have been accepted within a certain tolerance δ (i.e., $\delta = \pm 2cm$ for *height*) in order to ensure the availability of counterfactuals. Obviously, the smaller the delta, the greater is the probability that the algorithm will not return a counterfactual (i.e., lower availability), especially as the number of non-modifiable variables increases.

7.2.2.2 Results

Table 7.2 lists the properties of the sets of counterfactuals (see Section 7.1.2 for the definition) obtained for each different class of factuals $\mathbf{F}_i$. The discriminative power, for the different classes appears to be high, that is, above 95%, as shown in Table 7.2. This indicates that counterfactuals, although searched at a minimum distance,

Table 7.2: Availability (%), similarity(%), and discriminative power (mean% and C.I.%) of counterfactuals generated from FIFA dataset, for different factuals classes.

	FIFA			
Factual Class	MF	DE	FO	GK
C1 Class	DE	MF	MF	MF
Availability	100.00%	100.00%	100.00%	100.00%
Similarity (Mean)	21.73%	21.38%	21.39%	40.14%
Similarity (C.I.)	13.49%	12.74%	13.48%	35.80%
	29.96%	30.02%	29.31%	44.48%
C2 Class	FO	FO	DE	DE
Availability	100.00%	100.00%	100.00%	100.00%
Similarity (Mean)	23.35%	24.05%	24.34%	38.21%
Similarity (C.I.)	15.80%	16.94%	16.65%	34.11%
	30.89%	31.17%	32.04%	42.31%
C3 Class	GK	GK	GK	FO
Availability	100.00%	100.00%	100.00%	100.00%
Similarity (Mean)	40.13%	36.66%	37.60%	41.48%
Similarity (C.I.)	30.65%	27.71%	28.45%	36.95%
	49.61%	45.62%	46.75%	46.01%
Discriminative Power	95.58%	98.27%	98.89%	99.84%

Table 7.3: Classification performance: IRIS and Stellar datasets.

	IRIS	Stellar classification
$\mathbf{ACC_{tr}}$	95.24%	93.83 %
$\mathbf{OUT_{tr}}$	0.00%	0.01%
$\mathbf{ACC_{ts}}$	97.78 %	92.11 %
$\mathbf{OUT_{ts}}$	0.00%	0.02%
Macro F1-SCORE$_{\mathbf{ts}}$	97.78 %	94.18%
Cohen's Kappa$_{\mathbf{ts}}$	0.97	0.88

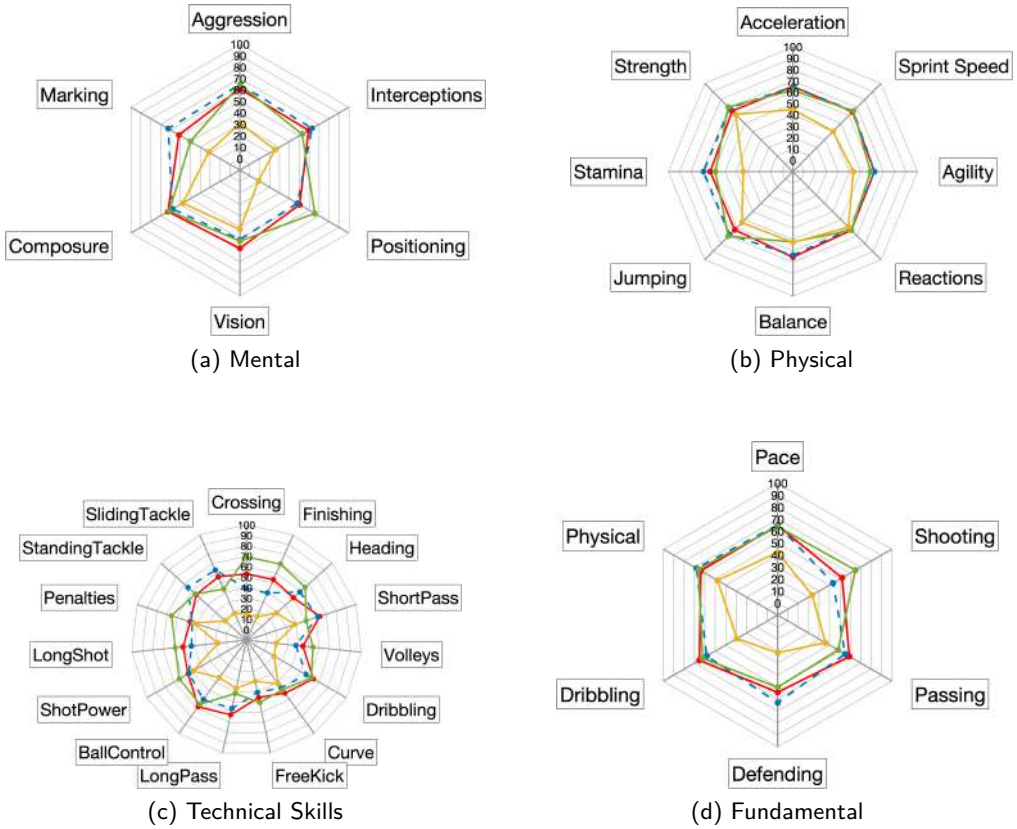


Figure 7.4: Each spiderplot represents the variation of the average of the factuals (dashed line) and counterfactuals (solid line) for DE class for each attribute category (Mental, Physical, Technical Skills and Fundamental). The value scale ranges from 0 to 100, and the output classes colors are the same as those used in Figure 7.3 (MF: red, DE: blue, FO: green, and GK: yellow).

are easily distinguishable from points belonging to the factual class. The highest discriminative power is computed with factuals belonging to the GK class, which, as previously mentioned, has more peculiar characteristics than the others. The algorithm successfully returned all counterfactuals (100% availability), demonstrating a sufficiently dense sampling of the SVDD regions. Lastly, similarity values are also satisfactory, with average values between 21% and 42%, depending on the factual class.

7.2.2.3 Knowledge extraction

The goal of the analysis is to identify which types of players are most characterized in their role and how different training plans can help specialize in a different role. For example, Fig. 7.4 analyzes the behaviour of the DE role, showing a spiderplot for each attribute category. It should be noted that the GK class differs significantly

Table 7.4: Availability (%), similarity(%), and discriminative power (mean% and C.I.%) of counterfactuals generated from IRIS and Stellar Classification datasets, for different factuals classes.

	IRIS			Stellar classification		
Factual Class	1	2	3	1	2	3
C1 Class	2	1	1	2	1	1
Availability	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Similarity (Mean)	33.93%	28.77%	49.93%	39.14%	16.15%	14.91%
Similarity (C.I.)	27.80%	16.89%	38.72%	18.79%	3.72%	2.50%
	40.07%	40.66%	61.14%	59.49%	28.58%	27.33%
C2 Class	3	3	2	3	3	2
Availability	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Similarity (Mean)	39.93%	11.83%	19.19%	14.78%	17.40%	38.68%
Similarity (C.I.)	33.92%	1.38%	9.13%	3.25%	6.29%	19.61%
	45.95%	22.29%	29.25%	26.31%	28.51%	57.76%
Discriminative Power	100.00%	82.91%	91.99%	95.09%	98.16%	98.10%

from the other classes. This is not surprising, since GK role requires different skills compared to the other roles. Concerning mental attributes, DE shows higher marking abilities than MF and FO. Moreover, DE positioning ability is similar to that of MF but remarkably lower than that of FO, whereas interceptions capabilities of DE are slightly higher than those of MF and FO. The remaining mental abilities present comparable values among DE, FO, and MF players. Physical attributes, instead, remain barely unchanged when considering DE, FO, and MF players. The only exception is the fact that DE and MF have on average greater balance than FO. Technical skills present different distributions when focusing on different classes of footballers. For example, DE short passing and long passing abilities are similar to those of MF and significantly higher than those of FO. Moreover, DE has higher values for both standing and sliding tackles than MF and FO. Intuitively, DE possesses worse abilities than FO when considering attributes strictly related to the attack phase including shot power, long shot, penalties, crossing, and finishing. Lastly, regarding the six fundamental attributes, on average DE, FO, and MF present comparable abilities in terms of pace, physical and dribbling abilities. Intuitively, DE players have higher defending abilities w.r.t MF and FO, and passing abilities intermediate between those of FO and MF. Reasonably, shooting capabilities are slightly lower than those of MF and strongly lower than those of a FO. After similar analysis of FO and MF spiderplots[4], the following conclusion arises. Workouts should be common on most abilities and strongly differentiated in target roles. For example, DE should focus on tackles and interceptions, FO on shooting and finishing, MF on passing. Other attributes, such as physical, aggressiveness, and dribbling, does not impact the specialization. Although such a conclusion may

[4]The corresponding spiderplots are available in: GitHub `https://github.com/AlbiCarle/MUCH/blob/main/SpiderImages/FIFA_SpiderPlots.pdf`.

appear intuitive, it may be of extreme interest to help experts (tactical and athletic coaches) in the selection of the target variables.

7.2.3 Characterization on Additional Datasets

This section discusses the performance of the proposed approach on a set of frequently referenced multi-class open source datasets, including the IRIS dataset [5] and the Stellar Classification Dataset - SDSS17 dataset [6]. These experiments help demonstrate that the approach can potentially scale well to tabular datasets of different size and different nature (i.e., physical measurements in the IRIS and SDSS17 datasets vs simulated play in the FIFA dataset).
The IRIS dataset consists of 150 observations related to peculiar characteristics of three different iris species (i.e., *Setosa*-1, *Versicolor*-2, and *Virginica*-3). Data records are equally balanced in terms of classes and records of the *Setosa* species are linearly separable from the others.
The Stellar Classification dataset includes 100,000 records of 3 type of objects (i.e., *galaxy*-1, *star*-2 or *quasar*-3) described by different spectral characteristics. Every observation consists of 17 input features, however only a subset of 10 features was considered in this experiment. Data records are equally balanced in terms of classes and records of the *Setosa* species are linearly separable from the others. Both datasets were split in training (70%) and test set (30%). Table 7.3 shows the training and test classification performance obtained by applying the MC-SVDD model, as presented in Section 6.1. Specifically, the classification performance is summarized in terms of accuracy and percentage of unclassified points on both training and test sets, macro-averaged F1-score and Cohen's Kappa on the test set. Table 7.4 shows the main properties of the set of counterfactuals obtained applying the method presented in Section 7.1 to the two state-of-the art datasets. Since class 1 in the IRIS dataset is linearly separable from the other 2 classes, counterfactuals belonging to classes 2 and 3 are very easily distinguishable from class 1 points. Indeed, the discriminative power for factual class 1 is 100% for both classes of counterfactuals.

7.3 Final Considerations

This work aims to formalize a multi-class generalization of an SVDD (MC-SVDD) and extract a set of counterfactual explanations from the classification results using a multi-class extension (MUCH) of a previously proposed counterfactuals explainer[159]. In order to be considered meaningful, a counterfactual should not only achieve the desired outcome minimizing the variation, but it should also be feasible, actionable, and retrieved fast enough. Experiments on three diverse datasets demonstrate that MC-SVDD is accurate in enclosing different classes of data points, with a negligible percentage of unclassified points. The use of a one-shot approach

[5] Retrieved [December 2022] from `https://www.kaggle.com/datasets/uciml/iris`

[6] fedesoriano. (January 2022). Stellar Classification Dataset - SDSS17. Retrieved [December 2022] from `https://www.kaggle.com/fedesoriano/stellar-classification-dataset-sdss17`

allows us to directly account for relationships and intersections between classes that would have been disregarded by considering multiple binary classifiers. Moreover, MUCH demonstrated satisfactory performance in terms of availability, similarity and discriminative power of the generated counterfactual explanations. The proposed MUCH approach has been applied starting from classification regions extracted from MC-SVDD, but it is in principle applicable to data regions derived from any machine learning method, like for example K-Nearest-Neighbors or rule based method. Future studies should therefore focus on proving that the MUCH generator is model-agnostic. Counterfactual explanations are minimal variations in input features that change the output class. This technique allows us to investigate the changes needed to move from the original class to a desired target class, as shown in Section 7.1. Similarly, in cases where it makes no sense to talk about passing between classes, counterfactuals can be used to characterize a dataset through the analysis of the peculiar characteristics that differentiate one class from another, as shown in Section 7.2.3. Three datasets have been shown as an example, but obviously the presented approach can be applied in several domains, such as the medical one, for example to study the impact of certain risk factors on the development of one or more diseases and subsequent preventive strategies. Future studies will focus in this direction. Moreover, the presented method should be further extended to handle different kinds of data, for example text, including textual explanations.

Part III

Real World Applications

Chapter 8

XAI against Adversarial ML

Machine learning (ML) algorithms are nowadays widely adopted in different contexts to perform autonomous decisions and predictions. Due to the high volume of data shared in the recent years, ML algorithms are more accurate and reliable since training and testing phases are more precise. An important concept to analyze when defining ML algorithms concerns adversarial machine learning attacks. These attacks aim to create manipulated datasets to mislead ML algorithm decisions. In this work, I and my team proposed new approaches able to detect and mitigate malicious adversarial machine learning attacks against a ML system. In particular, we investigated the Carlini-Wagner (CW), the fast gradient sign method (FGSM) and the Jacobian based saliency map (JSMA) attacks.
The aim of this work was to exploit detection algorithms as countermeasures to these attacks. Initially, we performed some tests by using canonical ML algorithms with a hyperparameters optimization to improve metrics. Then, we adopt original reliable AI algorithms, either based on eXplainable AI (Logic Learning Machine) or Support Vector Data Description (SVDD). The obtained results show how the classical algorithms may fail to identify an adversarial attack, while the reliable AI methodologies are more prone to correctly detect a possible adversarial machine learning attack.
The evaluation of the proposed methodology was carried out in terms of good balance between FPR and FNR on real world application datasets: Domain Name System (DNS) tunneling, Vehicle Platooning and Remaining Useful Life (RUL). In addition, a statistical analysis was performed to improve the robustness of the trained models, including evaluating their performance in terms of runtime and memory consumption.

Specifically, in this Chapter of my Thesis I report only the results achieved with the algorithms and methods developed during my research and exposed in the previous chapter of this thesis. The original work concerned more detailed and additional algorithms that I did not developed from myself. So I suggest the interested reader to delve in detatils with the full article published on IEEE Access, which link can be find in the **Publications** section.

8.1 Introduction

Machine learning (ML) has become an increasingly used technology in every aspect of our lives. It is adopted for image classification [71], to prevent health diseases [152], in cyber-security to detect cyber-attacks [127, 54], in the new industrial era (called industry 4.0) [77] or in other fields. It has a significant impact on daily activities and the use of these algorithms aims to improve daily life by offering services and applications capable of making optimal autonomous decisions. Obviously, the huge adoption of these algorithms is due to the large amount of data produced by the birth of emerging technologies such as the Internet of Things, smartwatches and smartphones. The extensive use of these algorithms and approaches has obviously also brought a benefit to the algorithms themselves as they have been more studied and applied, obtaining an improvement in performance, reliability, precision and calculation times.
Given the great use of ML algorithms, possible attacks on these systems have arisen in recent years. In particular, they are called adversarial machine learning. The main scope of these attacks is to inject malicious data (perturbed by an attacker starting from legitimate data) with the aim of making the algorithm misclassify or lower its accuracy [65]. The initial concept of the adversarial machine learning attack was focused on a misclassification of images [130] then it is moved to other fields such as in intrusion detection systems [149]. The detection phase of these attacks on ML algorithms is, to date, an important challenge in the research world as it is complex to identify malicious datasets as adversarial machine learning attacks use minimal global perturbations that make identification complex and challenging.

In this work, we proposed a new approach to identify an adversarial machine learning attack against a ML algorithm. As many attacks [68] and defensive approaches [64] are consolidated in the image analysis context, the topic is urgent for the more general framework of data analysis. In image settings, defensive techniques are strictly built around the sensitivity analysis of the functional cost of deep learning classifiers [64, 180]. They may thus result inapplicable to other kinds of classifiers in more general data analytics contexts.
Specifically, for this work I adopted the reliable SVDD algorithms described in the above sections of my Thesis. So the approach to contrast adversarial ML attacks was a mixture of black box and white box algorithms, to enhance, over the purpose of the detection, its explainability.

8.2 Adversarial Machine Learning

The concept of adversarial machine learning has been widely studied in the scientific literature in the last years. The inherent impact on the way to AI certification is becoming an urgent matter as well. In particular, poisoning attacks are defined as able to corrupt the training data so as to contaminate the ML model generated in the training phase, thus altering predictions on new data.

Technically speaking, [76] proposes an overview of the possible adversarial attacks to exploit the CIA (confidentiality, integrity and availability) requirements with a focus on a poisoning attack against images. Also [84] discusses all the possible adversarial attacks in a specific cyber warfare with a focus on possible privacy aspects. Then [109] offers a broad overview of the most widely used and efficient methodologies for dealing with adversary attacks in AI fields.

[99, 122, 124] instead analyze the issues that these attacks can lead to such as incorrect classifications or predictions in the medical field where an algorithm error may not identify a serious disease.

The adversarial machine learning concept is also considered in the malware detection approach where ML algorithms are adopted to detect a malicious mobile apps [80, 89]. This is a critical topic since smartphones contain sensitive information and a malware could retrieve these data, a correct classification aim to protect users from this threats [98, 137].

Another field aimed by adversarial attack is related to speech recognition. [90, 78] discusses the robustness of neural networks, adopted to speech recognition, to possible adversarial attacks. Authors demonstrate weaknesses of the speech algorithm on these attacks.

A critical context where ML algorithms are widely adopted is related to the Internet of Things (IoT). [111, 102] demonstrate how an adversarial attack could cause an alarm in case of fake detection of a cyber-attack against IoT devices. [119] discusses an adversarial machine learning attack by using a partial-model attack in order to manipulate the data fusion/aggregation process of IoT. Scope of this work is to lead the algorithm to make a wrong decision with respect to the input data of the IoT sensors.

Also, in [133], a detection approach of adversarial machine learning attacks is reported and presented. In this work, authors adopted canonical ML approaches to detect two adversarial attacks on a single dataset. Comparing with our work, we evaluated three adversarial attacks with canonical algorithms, innovative SVDD and XAI-based reliable approaches on three different datasets.

These are some examples of possible adversarial attacks against ML algorithms adopted in different contexts. Due to the criticality of this topic, we decided to work on a new approach to detect possible adversarial machine learning attacks by defining new approaches based on reliable AI through native eXplainable AI and SVDD approaches. As obtained results, the proposed algorithms are able to detect adversarial machine learning inference by obtaining a good balance between FPR and FNR. The results obtained demonstrate that the approach through reliable AI is more efficient than classic algorithms, also trained with a hyperparameter optimization. The proposed approach will be deeply discussed in the next sections and the results obtained will be detailed reported in order to demonstrate the efficiency and accuracy of the proposed algorithms and approach. We also evaluated our approach on three different datasets focused on different contexts: an intrusion detection (DNS), a collision avoidance (platooning) and a predictive maintenance (RUL) scenarios. In the next sections, we will detail the adopted approaches and

the obtained results.

8.3 Work concept

8.3.1 Principle behind adversarial

ML and artificial intelligence (AI) algorithms have been applied to many and different contexts in recent years, from the healthcare world to intrusion detection systems in the field of IoT security. Often, ML and AI models are trained using data retrieved from the environment to classify the different classes and make decisions based on the context in which they are applied. However, the trained models which support such systems may also be subject to attacks and thus introduce a new attack vector. Attacks that target the ML models are known as adversarial machine learning. The main aim of these attacks is to exploit the weaknesses of the trained model by manipulating and crafting data by starting from the real one. These perturbations increase the confusion in the decision model since ML algorithms are trained with different data. The perturbations performed by the adversarial machine learning attacks aim to be minimal to fool the model without an obvious change in the data used. Furthermore, another possible target of these attacks is to have the data misclassified in order, for example, to execute a cyber-attack on a system and classify it as legitimate. Although the concept of adversarial ML has been introduced in the field of images, in recent years several research works have dealt with introducing this concept in other contexts such as IoT [111], malware [98] or web applications [117].

8.3.2 Detection

We considered the following attacks: Carlini-Wagner, the Fast Gradient Sign method and the Jacobian based saliency map. In order to detect an adversarial attack against a victim ML algorithm, we decided to follow this approach: train a further ML binary classifier, by combining legitimate and adversarial data. The detection classifier is designed to identify as many attacks as possible, thus minimizing the False Positive Rate (FPR). In this way, more legitimate data may be misclassified as malicious (increase of false negatives), but a good compromise is seeked anyway under the adopted Reliable AI. After creating the combined dataset, we initially evaluate canonical ML algorithms, including decision tree, random forest, k-nearest neighbors (knn), gradient boost, support vector machine (svm) and logistic regression, with hyperparameters optimization to improve the detection performance.
We chose the mentioned algorithms as they are among the most used ones for binary classification problems in recent machine learning literature [150].
Then, Reliable AI is applied and compared with canonical ML. The workflow is shown in figure 8.1.
Our approach introduces a defensive technique through robustness enhancement outside the main training model, which is designed for the target application, e.g., visual landing, predictive maintenance, see, e.g., the Annex 2 of the EASA doc

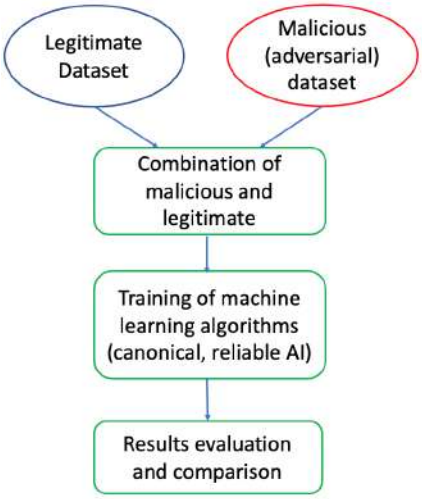


Figure 8.1: Concept approach to detect an adversarial machine learning attack. Datasets are represented by circles (blue for legitimate and red for malicious), while actions are represented by squares (green color).

[179]. That means the detection is still made by ML, but through another model that works in parallel with the main one and understands if the inputs provided to the main model are corrupted by adversarial distortions.

8.3.3 Target applications

The proposed activities are tested and evaluated on three different datasets: the first one is focused on network security (in particular, a DNS tunneling communication), the second one focused on vehicle platooning and the third one is a benchmark in predictive maintenance, consisting in Remaining Useful Life (RUL) estimation. The dataset relating to DNS tunneling is simpler as the legitimate and malicious data are divided into more distinct zones than with platooning and RUL, i.e., there are less overlaps of the two classes (legitimate and adversarial). In the platooning dataset, on the other hand, a strong superposition of points between the two classes makes the detection a hard task. Finally, in the RUL estimation original problem, the healthy and fault classes are quite well separated and we will investigate how the different proposed attacks impact on this base performance. In this way, we evaluated the proposed approach on increasing scenarios complexity. More information about the datasets is reported in the following.

Some more words are necessary for the assumptions made for the attacker and detection systems. This is the subject of the following two subsections.

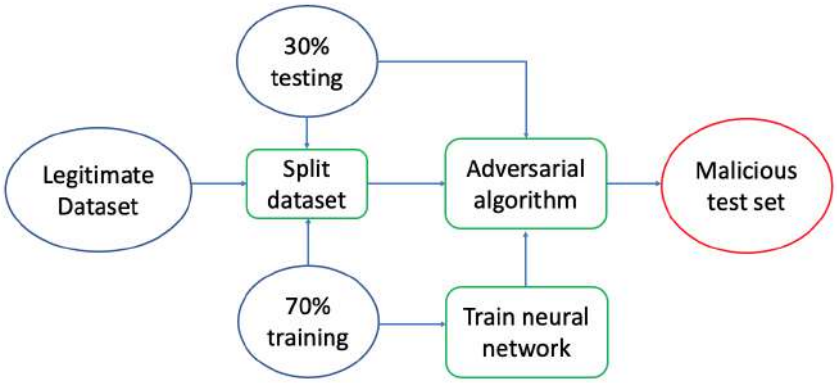


Figure 8.2: Concept approach to execute an adversarial machine learning attack. Datasets are represented by circles (blue for legitimate and red for malicious), while actions are represented by squares (green color).

8.3.4 Attacker assumption

During the adversarial attacks generation, a ML algorithm is required as victim of these attacks. Assuming that an attacker does not know the algorithms of a detection system, in this work we have decided to use a neural network as a victim of the various adversarial machine learning attacks. In particular, we decided to implement a neural network composed of 3 layers with the following numbers of nodes: 512, 256, 128 and a last layer as output. The model is trained with ReLu activation function for the hidden layers, a sigmoid function for the output layer, an Adam optimizer with learning rate set to $1.0e-5$, 300 epochs and batch size set to 16. During the training of the neural network, the accuracy was stably around 95 %. The workflow of the attack creation is reported in Figure 8.2. The original (legitimate) dataset is split into training and test portions; the victim neural network is then trained on training data and exploited by the adversarial attacks algorithms, that manipulate the test set in such a way to make that network misclassify data. This ends up in a malicious test set, which is then combined to legitimate data, as we better detail in the next.

8.3.5 Detection assumption

As for ML algorithms used for a specific use case in analysis, there are several considerations to take into account. Often, ML systems are trained using the data present in the system, to carry out classifications or forecasts, where the aim for these systems is to obtain high accuracy and precision metrics without considering adversarial attacks on the ML system. With this approach, an adversarial machine learning attack would be very successful as the algorithms would not be able to identify it. In this paper, we decided to consider possible adversarial machine learning attacks during the algorithm training phase. The classification/prediction system was trained with the different types of adversarial attacks to identify a possible at-

tack on the system. In this way, some legitimate data will be classified as malicious but the aim is to identify a possible attack by sacrificing possible legitimate values.

8.4 Adversarial attacks considered

In this section, we report a description of the adversarial machine learning algorithms considered in our work. In particular, we selected the Carlini-Wagner (CW), Fast Gradient Sign Method (FGSM) and Jacobian based Saliency Map attack (JSMA). The tool adopted to generate the adversarial machine learning attacks is the Adversarial Robustness Toolbox (ART) [94]. In the following formulas, we considered x as the legitimate input dataset while $\hat{x}$ as the adversarial dataset produced during the adversarial attacks, usually considered as $\hat{x} = x + \delta$ where $\delta \in [-1, 1]$ (data is supposed normalized in $[0, 1]$ [68]) is the perturbation of the attack.

8.4.1 Fast Gradient Sign Method

The Fast Gradient Sign Method (FGSM) attack was introduced by Goodfellow in the 2015 [48]. The malicious test set is generated by using the following equation:

$$\hat{x} = x - \varepsilon * sign(\nabla loss_{F,t}(x)) \tag{8.1}$$

In the above equation (8.1), x is the original input, t represents the target class and ε is the perturbation parameter, sufficiently small to be undetectable.
In the FGSM attack, a loss function is implemented to elaborate the input data to minimize the loss function. The attack proposed is able to misclassify the output of ML algorithms. This attack is tested and evaluated on different models to demonstrate its efficacy [48, 92, 63]. The main purpose of the FGSM attack is to be faster in the generation of adversarial test set at the expense of an optimal search for the best dataset in terms of perturbations [68]. This is considered the most efficient adversarial attack in terms of computing time and resources.

8.4.2 Jacobian based Saliency Map

Another adversarial attack considered in this work is the Jacobian Based Saliency Map (JSMA); it was introduced in 2016 by Papernot [65]. The concept is based on a simple assumption: understand how inputs affect outputs by modifying samples through the most influential features and tune them to achieve the most subtle, yet detrimental, effect on classification. The saliency map defines the gradients of the output over the input in canonical deep learning structures and it may drive comprehension, via visualization, of the image processing at each layer of the neural chain. The ranking of the values of the saliency map over the feature samples gives feature ranking. The JSMA process iteratively exploits such a feature ranking: the input is perturbed until a misclassification in the target class is achieved. If the desired misclassification is not reached, the JSMA inserts a new feature in the perturbation and tries to misclassify again.

It minimizes the L0 norm in order to produce the adversarial $\hat{x}$ by perturbing the feature with the highest impact [104], which is derived from the Jacobian matrix defined as follows. Given the i-th component as input and the j-th component as the derivative of the class [65]:

$$J_F(x) = \frac{\delta F(x)}{\delta x} = \left[\frac{\delta j(x)}{\delta x_i}\right]_{ixj} \tag{8.2}$$

8.4.3 Carlini-Wagner

The Carlini-Wagner (CW) attack is available in three different versions aimed to obtain low distortion for this metrics [68]: the first aims to minimize the L_2 norm, the second the L_0 norm and finally the third the L_∞ norm. In this work, we adopted the version focused on the L_2 norm.

$$\begin{aligned} &\arg\min_{\delta} \; D(x, \hat{x}) \\ &\text{such that } C(\hat{x}) \neq C^*(x) \\ &\text{where } \hat{x} \in [0,1]^n \end{aligned} \tag{8.3}$$

where $\hat{x}$ is considered as $x+\delta$ and C is the classifier considered; $C^*(x)$ is the optimal classification of x and $D(\cdot)$ is a proper distance metric. The CW attack is also robust against defensive distillation [64] or other detection mechanisms [68]. Until now, this attack is considered as the most powerful among the existing attacks. Obviously, since the generation of malicious adversarial data is very accurate, computational times can be very long.
In this section, we initially present the datasets considered in the proposed work. Then, we discuss a first approach to detect adversarial machine learning attacks by using classical algorithms with hyperparameters optimization to improve performance metrics. Then, we move to the reliable SVDD approach, also combined with rules extraction. As metrics to measure the detection of adversarial attacks, we adopt the confusion matrices in order to evaluate the correct classification of legitimate and malicious data.

8.5 Clarifying example

8.5.1 Datasets

The used datasets represent two challenging scenarios for detection even without the adversarial component[1]. The first one deals with covert channel detection in cybersecurity [54]; more specifically, the aim is detecting the presence of Domain Name Server intruders by an aggregation-based monitoring that avoids packet inspection,

[1]The adopted datasets, both the original legitimate and the attacked ones, are available as open-source in the following repository:`https://www.kaggle.com/datasets/cnrieiit/adversarial-machine-learning-dataset`.

in the presence of silent intruders and quick statistical fingerprints generation. By modulating the quantity of anomalous packets in the server, we are able to modulate the difficulty of the inherent supervised learning solution via canonical classification schemes (Bayes decision theory, neural networks). More specifically, let q and a be the packet sizes of a query and the corresponding answer, respectively (what answer is related to a specific query can be understood from the packet identifier) and Dt the time-interval intercurring between them. The information vector is composed of the statistics (mean, variance, skewness and kurtosis) of q, a and Dt for a total number of 12 input features:

$$\mathbf{I} = [m_A, m_Q, m_{Dt}, v_A, v_Q, v_{Dt}, s_A, s_Q, s_{Dt}, k_A, k_Q, k_{Dt}]$$

The corresponding vectors are: $\mathbf{m}$, $\boldsymbol{\sigma}$, $\mathbf{s}$, $\mathbf{k}$. High-order statistics give a quantitative indication of the asymmetry (skewness) and heaviness of tails (kurtosis) of a probability distribution, they help improve detection inference.
The second dataset addresses collision prediction in vehicle platooning [93], which is widely considered one of the most challenging problems in smart mobility scenarios. It consists of a group of vehicles interconnected via wireless that travel autonomously, based on the widespread Cooperative Adaptive Cruise Control (CACC) technology [58]; the aim is to find a compromise between performance (e.g., maximize speed and minimize reciprocal distance, thus minimizing air drag resistance and fuel consumption, too) and safety (no collision, even in the presence of anomalous events, such as sudden brakes [93]).
The behavior of the platooning system is synthesised by the following vector of features:

$$\mathbf{I} = [N, F_0, PER, d_0, v_0]$$

where N is the total number of vehicles of the platoon, F_0 is the braking force applied by the leader, PER is the probability of packet loss, and d_0 and v_0 are the mutual distance and speed between each pair of vehicles in the initial condition. The ML solution is based on a supervised classification task that maps current speed, distance, acceleration, weight of vehicles, as well as quality of service of the communication channel, into a potential collision into the near future. As shown later, the adversarial component makes a detrimental impact on the chances to find such a mapping.
Another realistic application scenario is represented by the Turbofan Engine Degradation Simulation Data Set, made available by NASA [178]. It is an important benchmark in predictive maintenance, since it deals with damage propagation modeling for aircraft engines. The repository contains four different sets of data, called FD001, FD002, FD003 and FD004 [2], corresponding to simulations under different combinations of operational conditions and fault modes. Our analysis here is based on FD001 only. Different functional parameters of aircraft gas turbine engines are collected by sensors over time and describe the trajectory of the system (more information on all the available measurements can be found in the original

[2] `https://shorturl.at/ewGM5`

publication[31]). Features are then extracted, by computing the mean m, variance v, skewness s and kurtosis k for each parameter raw time-series, over a moving time window (observation horizon) of fixed size, obtaining samples making up what we call RUL dataset. The goal is to recognize those trajectories that may result in fault states, based on the extracted features. This implies the definition of the Remaining Useful Life (RUL) variable, which represents how much time is left before a fault occurs. In practice, one would want to understand which conditions are inherent to imminent faults of the engine. The problem can be solved via a ML classification task, where the RUL constitutes the output class. In the original dataset, the RUL class assumes three values: healthy (RUL>150), critical (50≤RUL≤150), and faulty (RUL<50). For our application, we further elaborated the data by reducing its dimensions to the following features:

$$\mathbf{I} = [s_{os_2}, m_{N_c}, v_{N_c}, v_{phi}, m_{htBleed}, s_{htBleed}, m_{W31}]$$

The choice on these variables was done by evaluating after a feature evaluation on the FD001 dataset. Moreover, due the high under-sampling of faulty class and for consistency with the other two applications, we decided to merge the critical and faulty samples into a single faulty class. Hence, the problem becomes a binary classification between healthy (RUL>150) and faulty (RUL≤150). After the attacks generation, following the approaches summarized in Figures 8.1 and 8.2, each described dataset is be merged (as legitimate points) to the malicious (attacked) test set.

8.5.2 Canonical supervised learning and hyperparameter optimization

In order to provide a first possible protection from the adversarial machine learning attacks, we focused on the adoption of classic ML algorithms. This approach is adopted to validate if classic ML algorithms are able to correctly classify possible adversarial attacks. For this reason, we implemented different algorithms such as decision tree, gradient boost, K-nearest neighbors, logistic regression, random forest and a support vector machine. The dataset, composed by legitimate and malicious rows, is splitted in 70% of training and 30% of testing. The algorithms were implemented through the Sklearn [40] library, an open source ML library for the Python programming language. The tests were performed with the same dataset and on the same machine to avoid differences in obtained results to guarantee consistency on the tests and results. Moreover, the number of rows of the legitimate and adversarial datasets are of the same order of measurement in order to have a balanced dataset since an unbalanced dataset could reports high values of metrics.
As presented previously, we tested the ML algorithms on three different datasets: DNS tunneling, platooning and RUL estimation. The results obtained are reported by using metrics extracted from confusion matrices, in particular we decided to report false positive rate (FPR), true positive rate (TPR), false negative rate (FNR) and true negative rate (TNR). All results are shown in Table 8.1, divided by algorithm, attack and dataset.

For the results on the DNS dataset, it is possible to note that, with default configurations of the algorithms, the detection of an adversarial machine learning attack is not achieved since most of the algorithms are not able to correct classify the malicious payload. Also analyzing the results for the platooning dataset, we can note that the confusion matrices report low values of correct classification. In particular, algorithms are not able to classify the attack as demonstrated by the high number of false positive rate and false negative rate. By focusing only on the correct classification of the attack, most of the algorithms classify the attack as legitimate. In particular, for the CW attack, only the SVM algorithm manages to classify it well enough, at the expense of many incorrect classifications of legitimate data. These results are also validated by the FPR table reported in Table 8.1 where all the algorithms have high value of FPR except for the SVM which has 0.03 but related to a wrong classification of the legitimate data, so this value is not considerable as good result. While considering the JSMA attack, all the algorithms are not able to classify with good performance the attack. Finally, regarding the FGSM, the random forest and decision tree obtained a good FPR value but the other algorithms are not able to perform a correct classification. These results actually demonstrate that the adversarial attacks are complex to identify since with minimum perturbations of the dataset, the behaviour of a ML algorithm is totally confused.
The performance of canonical ML methods on RUL dataset differs from DNS and platooning, since they perform bad on CW attack, managing to lower the FPR but at the cost of very high FNR values. In contrast, on JSMA and FGSM they generally perform well, with the surprising exception of SVM on FGSM attack, whose performance is the same as on CW (FPR=0, TPR=0).

		DNS				**Platooning**				**RUL**			
		FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR
Decision Tree	CW	0.13	0.29	0.87	0.71	0.43	0.55	0.57	0.45	0.23	0.03	0.77	0.97
	JSMA	0.48	0.80	0.52	0.20	0.45	0.87	0.55	0.13	0.00	1.00	1.00	0.00
	FGSM	0.04	0.27	0.96	0.73	0.18	0.86	0.82	0.14	0.00	1.00	1.00	0.00
Gradient boost	CW	0.49	0.99	0.51	0.01	0.58	0.69	0.41	0.31	0.001	0.00	0.999	1.00
	JSMA	0.50	0.99	0.50	0.01	0.35	0.88	0.65	0.12	0.00	1.00	1.00	0.00
	FGSM	0.02	0.16	0.98	0.84	0.23	0.91	0.77	0.09	0.0002	0.998	0.9998	0.002
KNN	CW	0.85	0.80	0.15	0.20	0.46	0.49	0.53	0.51	0.05	0.02	0.95	0.98
	JSMA	0.96	1.00	0.04	0.00	0.62	0.82	0.38	0.18	0.00	1.00	1.00	0.00
	FGSM	0.62	0.82	0.38	0.18	0.49	0.52	0.51	0.48	0.008	0.4105	0.99	0.59
Logistic regression	CW	0.50	1.00	0.50	0.00	0.59	0.64	0.4	0.36	0.00	0.00	1.00	1.00
	JSMA	0.36	1.00	0.64	0.00	0.51	0.91	0.49	0.09	0.00	1.00	1.00	0.00
	FGSM	0.03	0.93	0.97	0.07	0.48	0.63	0.52	0.37	0.01	0.53	0.99	0.47
Random forest	CW	0.32	0.70	0.68	0.30	0.48	0.62	0.51	0.38	0.17	0.002	0.83	0.998
	JSMA	0.50	0.99	0.50	0.01	0.41	0.87	0.58	0.13	0.00	1.00	1.00	0.00
	FGSM	0.03	0.13	0.97	0.87	0.18	0.91	0.82	0.09	0.00	1.00	1.00	0.00
SVM	CW	0.98	0.98	0.02	0.02	0.03	0.11	0.97	0.89	0.00	0.00	1.00	1.00
	JSMA	0.10	0.59	0.90	0.41	0.81	0.92	0.19	0.08	0.00	1.00	1.00	0.00
	FGSM	0.25	0.92	0.75	0.08	1	1	0	0	1.00	1.00	0.00	0.00

Table 8.1: **Canonical machine learning.** The table shows the performance statistics of canonical machine learning algorithms, divided by attack and dataset.

As possible to note from the above discussions, with default configurations of the algorithms, the detection of an adversarial machine learning attack is not achieved since most of the algorithms are not able to correctly classify the malicious payload. However, to carry out a correct classification, a detailed and specific configurations of the models must be tested and validated. For this reason, we decided to perform

		DNS				Platooning				RUL			
		FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR
Decision Tree	CW	0.50	1.00	0.50	0.00	0.43	0.55	0.57	0.45	0.00	0.00	1.00	1.00
	JSMA	0.25	1.00	0.75	0.00	0.23	0.84	0.77	0.16	0.00	1.00	1.00	0.00
	FGSM	0.50	1.00	0.50	0.00	0.13	0.85	0.87	0.15	0.0006	1.00	0.9994	0.00
Gradient boost	CW	0.48	1.00	0.52	0.00	0.44	0.59	0.56	0.41	0.00	0.00	1.00	1.00
	JSMA	0.50	1.00	0.50	0.00	0.33	0.88	0.67	0.12	0.00	1.00	1.00	0.00
	FGSM	0.03	0.36	0.97	0.64	0.12	0.92	0.88	0.08	0.00	1.00	1.00	0.00
KNN	CW	0.97	1.00	0.03	0.00	0.62	0.69	0.37	0.31	0.00	0.00	1.00	1.00
	JSMA	0.89	1.00	0.11	0.00	0.53	0.84	0.46	0.16	0.00	1.00	1.00	0.00
	FGSM	0.11	0.28	0.89	0.72	0.47	0.57	0.53	0.43	0.00	1.00	1.00	0.00
Logistic regression	CW	0.49	0.99	0.51	0.01	0.51	0.6	0.49	0.4	0.00	0.00	1.00	1.00
	JSMA	0.09	0.98	0.91	0.02	0.34	0.79	0.66	0.21	0.00	1.00	1.00	0.00
	FGSM	0.03	0.99	0.97	0.01	0.56	0.78	0.44	0.22	0.00	0.81	1.00	0.19
Random forest	CW	0.49	1.00	0.51	0.00	0.46	0.66	0.54	0.34	0.00	0.00	1.00	1.00
	JSMA	0.50	1.00	0.50	0.00	0.33	0.87	0.67	0.13	0.00	1.00	1.00	0.00
	FGSM	0.03	0.32	0.97	0.68	0.17	0.92	0.83	0.08	0.00	0.9992	1.00	0.0008
SVM	CW	0.39	0.65	0.61	0.35	0.63	0.85	0.37	0.15	0.00	0.00	1.00	1.00
	JSMA	0.09	0.98	0.91	0.02	0.24	0.69	0.76	0.31	0.00	1.00	1.00	0.00
	FGSM	0.15	0.95	0.85	0.05	0.69	0.89	0.31	0.11	0.00	0.00	1.00	1.00

Table 8.2: **Canonical machine learning optimized.** The table is the optimized version of algorithms presented in Table 8.1 in which the performance statistics of the optimized canonical machine learning algorithms divided by attack and dataset are shown.

a hyperparameter optimization of the ML algorithms in order to validate if the adversarial attacks can be correctly identified by tuning the models adopted to detect them. The hyperparameter optimization challenge is to select a set of optimal parameters for a ML algorithm to improve the evaluation metrics and the precision of a model.

In order to achieve these results, we adopted the hyperparameters optimization tool called Optuna [96]. Optuna formulates the hyperparameter optimization as a process of minimizing/maximizing an objective function that takes a set of parameters as input and returns its score. Once the parameters to be optimized have been defined, Optuna starts combining the different parameters and evaluating the algorithm to validate if a combination of the parameters leads to an algorithm improvement in terms of metrics. At the end of the parameter testing process, the optimal ML model returns the parameters that calculate the higher metrics. In our tests, to obtain efficient results and to test a good number of parameter combinations, 1000 parameter combinations with different values (chosen by Optuna according to a tool logic) were performed for each algorithm.

Once the optimal parameters for the algorithms were obtained, confusion matrices of each algorithms were calculated to validate if, with the hyperparameter optimization, the ML model is able to detect the adversarial attacks. Obtained results are reported in Table 8.2.

Regarding the DNS dataset, by analyzing the obtained results, the hyperparameter optimization improved the metrics. In particular, in the FGSM and the JSMA, most of the algorithms are able to correct classify the adversarial attacks while in the CW the metrics are still low since the CW attack is more complex than the others.

After the implementation of the hyperparameter optimization, an improvement of the metrics on the platooning dataset is obtained. For the CW attack, the metrics value for each algorithm is again high, so a classification of the attack is not suitable

for a real environment. In particular, the SVM is near the 63% of FPR due to a correct classification of the legitimate data. For the JSMA attack instead, the FPR metric is near to the 25%-30% which is again not good for classifying the attack. Finally, the FGSM attack obtained good results for the decision tree, gradient boost and the random forest with a FPR rate lower than 17% due to the fact that this attack is more simple to detect (since it requires minor time to be executed so it is not accurate).
The hyper-parameters optimization on RUL dataset was effective for the cases where the detection was already satisfactory, i.e. on the JSMA and FGSM; CW attack remains hardly detectable with any canonical method instead.

8.5.3 SafeSVDD

		DNS				Platooning				RUL			
		FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR
zeroFPRSVDD	CW	0.04	0.35	0.95	0.64	0.11	0.21	0.89	0.78	0.03	0.03	0.97	0.97
	JSMA	0.15	0.85	0.84	0.14	0.09	0.36	0.90	0.63	0	0.99	1	0.01
	FGSM	0.03	0.77	0.96	0.22	0.13	0.14	0.86	0.85	0.01	0.99	0.99	0.01
eXplainableSVDD	CW	0.23	0.35	0.76	0.64	0.34	0.34	0.65	0.65	0.44	0.73	0.55	0.26
	JSMA	0.28	0.53	0.71	0.46	0.34	0.30	0.65	0.69	0.50	0.57	0.50	0.43
	FGSM	0.28	0.28	0.71	0.71	0.35	0.33	0.64	0.66	0.57	0.55	0.43	0.45

Table 8.3: FPR, TPR, TNR and FNR for each dataset and attack with the Safe SVDD methods.

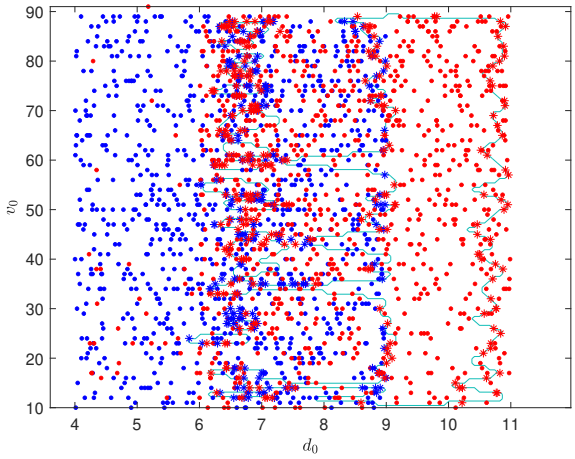


Figure 8.3: 2D graph of the "adversarial region" (the red points are the attacked ones) with d_0 (distance between cars) and v_0 (initial platooning speed) as input features of the JSMA-platooning dataset. The star points are the SVs of the description, coloured referring their specific label.

In order to improve the results obtained with classical ML algorithms showed in Section 8.5.2, our goal is to determine the largest region of parameters with no false positives (i.e. prediction of attack, but no attack in reality) using the algorithms

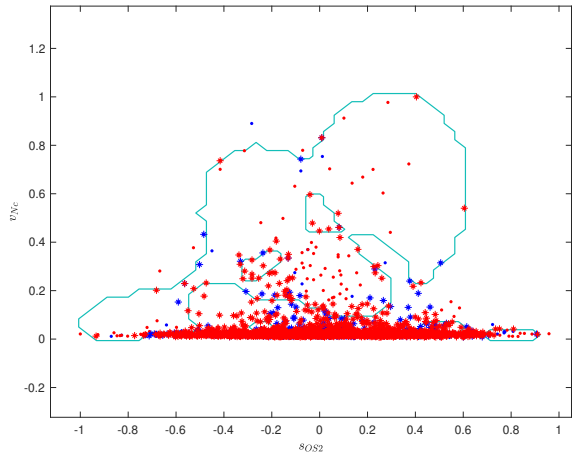


Figure 8.4: 2D graph of the "adversarial region" (the red points are the attacked ones) with s_{OS2} (Skewness of operational setting 2) and v_{Nc} (Variance of physical core speed) as input features of the CW-RUL dataset. The star points are the SVs of the description, coloured referring their specific label.

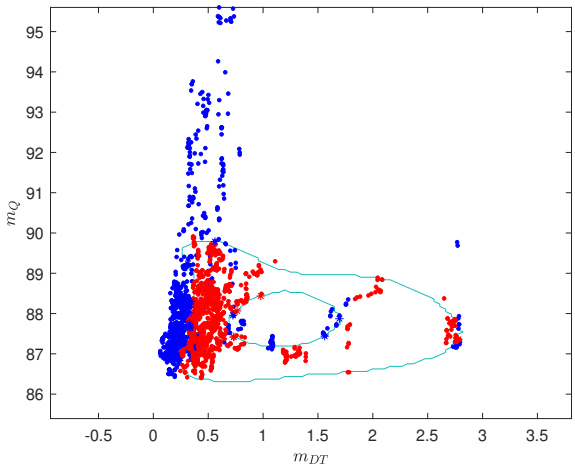


Figure 8.5: 2D graph of the "adversarial region" (the red points are the attacked ones) with m_{Dt} (average interarrival time between query and answer packet over 1000 sample) and m_Q (average size of query packet) as input features of the JSMA-DNS dataset. The star points are the SVs of the description, coloured referring their specific label.

proposed.

For the `zeroFPRSVDD` algorithm we set $C_1 = 1/(\nu_1 N_1)$, where $N_1 = \#\{y_i = +1\}$

and $\nu_1 = 0.01$ (i.e. we allow the acceptance of up to 1% of negative objects in the target class), $C_2 = 1/(\nu_2 N_2)$ where $N_2 = \#\{y_i = -1\}$ and $\nu_2 = 0.05$ (i.e. we allow up to 5% negative objects to be included in the SVDD region) and we used the RBF kernel with σ determined with cross-validation for all the three datasets and attacks. The results are shown in Table 8.3. Let's pay more attention on the FPR index since it is the one that explains most the aim of the Safe SVDD: recall you that the purpose of the Safe SVDD is to find the largest region with the lowest rate of negative points within it, so we are less interested in what happens outside the Safe SVDD region. The performances on the three datasets show results totally in line with the other methodology. In particular we can notice that the CW attack is the most difficult to detect, emphasizing the hypothesis that the CW attack is the attack that most distorts the output of the algorithm under attack.

Although for some attacks the safety regions determined are not very large, with this algorithm we are sure to find areas with very low misclassification error (tending to zero) in relation to the target class. In particular, when compared to SVM algorithm (to which the SVDD is closely related [10]) we can observe that the results have been improved.

Regarding the `eXplainable SVDD` algorithm, for each classification made via `zeroFPRSVDD` algorithm we extracted the set of intelligible rules and performed again the classification of the datasets. Results are reported in Table 8.3.

Not surprisingly, the performance of `eXplainableSVDD` is inferior to that of the other algorithm: it is the price to pay for extracting explainability from a black-box algorithm. With the explainable version of the safe SVDD, we try to approximate complex decision boundaries with rectangles, i.e., rules. Thus, to avoid exponential generation of rules to exactly describe decision boundaries (which would not be as explainable and useful to a potential user), it is preferable to admit a larger margin of error. This way you get fewer but more understandable rules.

In the case where safety regions are not operationally representative (as is also the case with canonical machine learning), it is necessary to admit that it is not possible to obtain zero statistical error. Therefore, it is better to allow the algorithms to have a higher probability of error (i.e., set the threshold on the number of FPRs higher) to still obtain the possibility of having a sufficiently significant data region in which to apply appropriate countermeasures.

As an example of the kind of rules generated by `eXplainable SVDD`, in the CW-DNS dataset we set $\varepsilon = 0.1432$, in the JSMA-platooning dataset we set $\varepsilon = 0.0184$ and in the FGSM-RUL dataset we set $\varepsilon = 0.0167$. Always referring to the three datasets in example, the first one highest-covering rule (i.e. the rule involving the largest

number of data points, (5.1)) for the class *attack* for CW-DNS dataset is

if $(30931149 < v_A \leq 166588766)$
and $(211 < v_Q \leq 2604)$ and $(3779 < v_{Dt} \leq 155832)$
and $(360 < s_{Dt} \leq 392)$ and $(52 < s_A \leq 326)$
and $(368 < k_{Dt} \leq 4874$ and $(29 < k_A \leq 328)$
then attack

the first three rules for JSMA-platooning dataset are

1. **if** $0.37 < PER \leq 0.89$ and $7.38 < d_0 \leq 7.59$
   and $54 < v_0 \leq 66$ **then attack**
2. **if** $N < 5$ and $0.36 < PER \leq 0.78$
   and $9.81 < d_0 <= 9.94$ **then attack**
3. **if** $N < 5$ and $F_0 < -2$ and $0.37 < PER \leq 0.53$
   and $7.26 < d_0 \leq 9.43$ and $29 < v_0 \leq 78$ **then attack**

and the first rule for FGSM-RUL dataset is

if $s_{os2} \leq 0.33$ and $m_{Nc} < 9060.24$
and $132.14 < v_{Nc} \leq 526.57$ and $0.04 < v_{phi} \leq 0.12$
and $38.22 < m_{W31} \leq 39.33$ **then attack**

As we were saying above, the fact that the rules are very intricate and that each rule involves almost all input parameters is because we are approximating the nonlinear form of SVDD with hyper-rectangles, i.e. rules. To ensure acceptable prediction confidence with these rules, a large amount of them is required: for the cases in example, JSMA-platooning and CW-DNS, the total number of rules generated are 751, 146 and 102 respectively. Moreover, having a high number of rules means having low coverage for each rule: this may suggest that, first, the task is very difficult but, second, that the regions developed by SVDD are widely and sporadically distributed inside the space of the input parameters.
These results show how SVDD can improve LLM algorithm for the detection of the attacked points in the datasets. Moreover, this procedure offers a simple and clear method for making SVDD explainable which is quite innovative with respect the well known methods for extracting rules from SVM [26, 34].

As a final remark, results obtained through canonical SVM with hyper-parameters optimization for DNS tunneling, under JSMA and FGSM, also reveal good detection ability, as well as all the canonical methods on RUL dataset, but, as briefly shown in the next section, the computational costs are demanding.
In order to carry on a statistical computational analysis, we evaluated the consumption of CPU and RSS memory (Resident Set Side, hence related to the memory

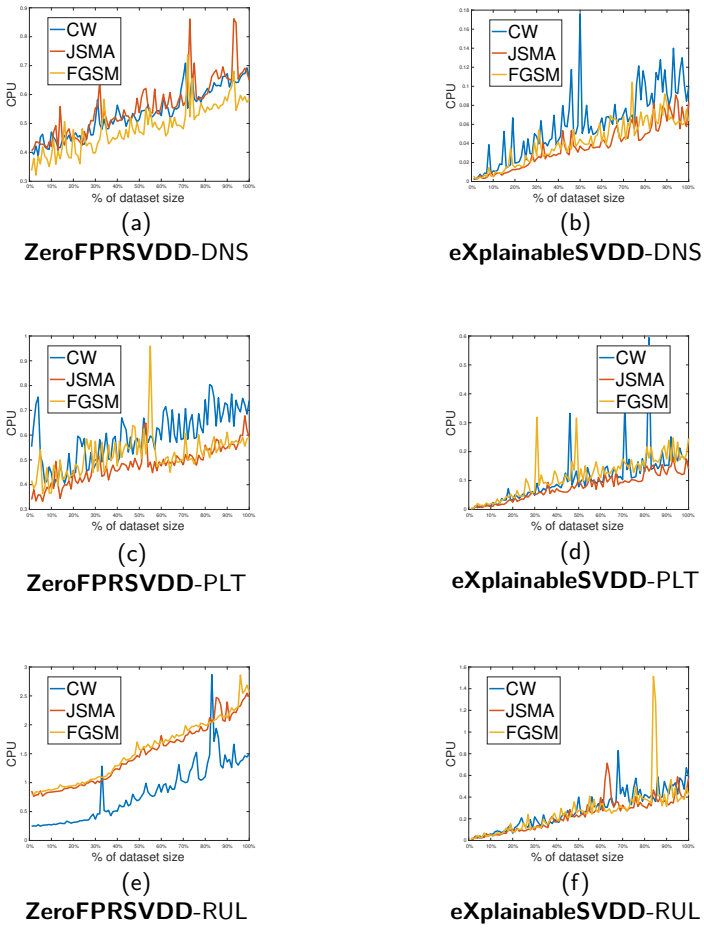


Figure 8.6: Graphs of the processing time (denoted as CPU in the figure) as the dataset and algorithm change: first row DNS dataset, second row Platooning dataset (denoted with PLT in the plot captions) , third row RUL dataset.

allocated to the process in RAM) for each dataset and attack. We split the test dataset into 100 datasets of increasing size, evaluating the algorithm on each dataset. What it is worth to note is that the computation of `zeroFPRSVDD` algorithm is quite demanding, since the algorithm works on MATLAB (R2021). Instead, computationally speaking, `eXplainableSVDD` algorithm performs better, since it works on Python3.

8.6 Conclusion comments

In this work, we investigated an innovative approach to detect adversarial machine learning attacks by comparing canonical ML algorithms with two innovative Reliable AI approaches focused on Support Vector Data Description (SVDD). In particular, we investigated three possible adversarial attacks, namely the Carlini-Wagner, the

		DNS		PLATOONING		RUL	
		CPU	RSS	CPU	RSS	CPU	RSS
zeroFPRSVDD	CW	0.54±0.08	981114.16±49663.31	0.59±0.10	991341.28±44006.02	0.82±0.49	1168571.40±233561.18
	JSMA	0.56±0.10	981126.20±50738.10	0.48±0.07	991242.68±40774.24	1.48±0.52	1430018.56±223946.24
	FGSM	0.49±0.08	977234.60±54004.73	0.52±0.08	991892.68±45294.48	1.54±0.53	1449057.04±223909.95
eXplainableSVDD	CW	0.06 ± 0.03	9151±2.05	0.10±0.08	9156±1.90	0.28± 0.17	9112±1.12
	JSMA	0.03±0.02	9152±1.37	0.07±0.04	9159±0.00	0.26±0.21	9112±0.00
	FGSM	0.04±0.02	9152±0.0	0.11±0.06	9156±0.79	0.24±0.15	9112±0.00

Table 8.4: Processing time (CPU) and memory consumption (RSS) for Safe SVDD methods tested on 100 test sets with increasing sizes (results reported as mean ± standard deviation)

Fast Gradient Sign Method and the Jacobian based saliency map. The proposed approach plans to generate malicious datasets (i.e. under attack by adversarial algorithms) on the defensive side to train the algorithms by combining a malicious dataset with the legitimate one. In this way, the algorithm is able to identify a possible attack certainly sacrificing legitimate data but, the basic idea of the work, provides for a classification of adversarial machine learning attacks.
Regarding the shortcomings of our methodology, the proposed detection framework is not yet complete for preventing adversarial attacks. The method works a posteriori, after the attack has been carried out, so it is not possible to apply countermeasures to prevent the attack. However, the information obtained from the detection phase can be exploited to identify or prevent subsequent attacks, since the method clearly defines the adversarial regions. It will be future work to study how to use our method to prevent the attacks or to apply suitable countermeasures.

Chapter 9

CE for Type 2 diabetes prevention

During my research, I and my team applied our algorithm to real world scenario in order to better provide the state-of-the-art with improved and well contextualized machine learning tools. In particular in this work, we focused on the prediction of type 2 diabetes via conterfuactual explanations that, despite the growing availability of artificial intelligence models forits prediction, there is still a lack of personalized approaches to quantify minimum viable changes in biomarkers that may help reduce the individual risk of developing disease.

In this thesis I reported a general overview of the analysis and results obtained effectively. The interested reader can find the link to the full paper in the **Publications** section.

9.1 Introduction

The aim of this work was to develop a new method, based on counterfactual explanations, to generate personalized recommendations to reduce the one-year risk of type 2 diabetes. Ten routinely collected biomarkers extracted from Electronic Medical Records of 2791 patients at low risk and 2791 patients at high risk of type 2 diabetes were analyzed. Two regions characterizing the two classes of patients were estimated using a Support Vector Data Description classifier. Counterfactual explanations (i.e., minimal changes in input features able to change the risk class) were generated for patients at high risk and evaluated using performance metrics (availability, validity, actionability, similarity, and discriminative power) and a qualitative survey administered to seven expert clinicians.

9.2 Methodology

The aim of this study was the development of a novel method based on counterfactual explanations to produce *personalized* minimum viable modifications of routinely measured biomarkers potentially able to reduce the risk of developing T2DM. We used the framework described in Chapter 6.2 in order to identify these modifications. Specifically, in [163], we applied the proposed method to characterize T2DM using an unbalanced set of 1857 subjects (428 diagnosed with T2DM and 1429 without the disease) and biomarkers derived from electronic medical records (EMRs). We demonstrated that the minimal variations in the input features associated with a change in the output class were coherent with the literature related to T2DM. Specifically, diabetic patients were on average associated with higher fasting blood sugar (FBS), higher body mass index (BMI), and lower high-density lipoproteins (HDL), compared to their non-diabetic counterfactuals. The method relied on the definition of two TC-SVDD classification regions named "T2DM" and "No T2DM" and on the generation of a set of counterfactuals that, being by definition at minimum distance, were located near the decision boundary of the "No T2DM" class. However, the method developed in [163] is nor readily applicable to diabetes prevention and risk reduction for the following reasons. First, the boundaries of the two regions are very close to each other and, as a result, the observed changes in biomarkers may not be able to decrease the risk of disease and, as such, may not be translated into practical preventive recommendations. In principle, larger changes may be obtained by using smaller regions to define the "No T2DM" class. For example, by reducing the false negative rate (FNR) of the TC-SVDD classifier, a smaller, more conservative, "No T2DM" region can be obtained and used to characterize patients without the disease that are inherently different from those in the "T2DM" region. Second, in [163], the counterfactuals were assessed only in terms of average differences and no human validation of the observed changes in biomarkers was performed. Last, in [163] we focused on characterization of patients already diagnosed with T2DM, rather than on the investigation of preventive recommendations on individuals at risk of developing T2DM in the future. For a clear identification of actionable counterfactuals able to reduce the risk of developing disease, a different dataset than the one used in [163] is needed.

The main contributions and advancements of the present study compared to previous literature and, particularly, compared to [171] and [163] are summarized in the followings:

- selection of a dataset including individual observations before the onset of T2DM to investigate which biomarkers and which change in biomarkers can help reduce the risk of developing T2DM;
- development of a novel methodology for the generation of *actionable* counterfactual explanations from numerical and categorical tabular data by varying only a subset of *controllable* features and constraining *non controllable* features

such as age and sex;

- generalization of the TC-SVDD classifier that defines the two regions of the output classes by controlling the FNR to modulate the risk associated with the "low" risk output class and obtain "more conservative" minimal changes towards a lower risk of developing T2DM;
- assessment of the proposed XAI framework through an *ad-hoc* survey delivered to medical experts, in line with the *Human agency and oversight* requirement of trustworthy AI;
- comparison of the newly proposed methodology with two state-of-the-art local XAI techniques.

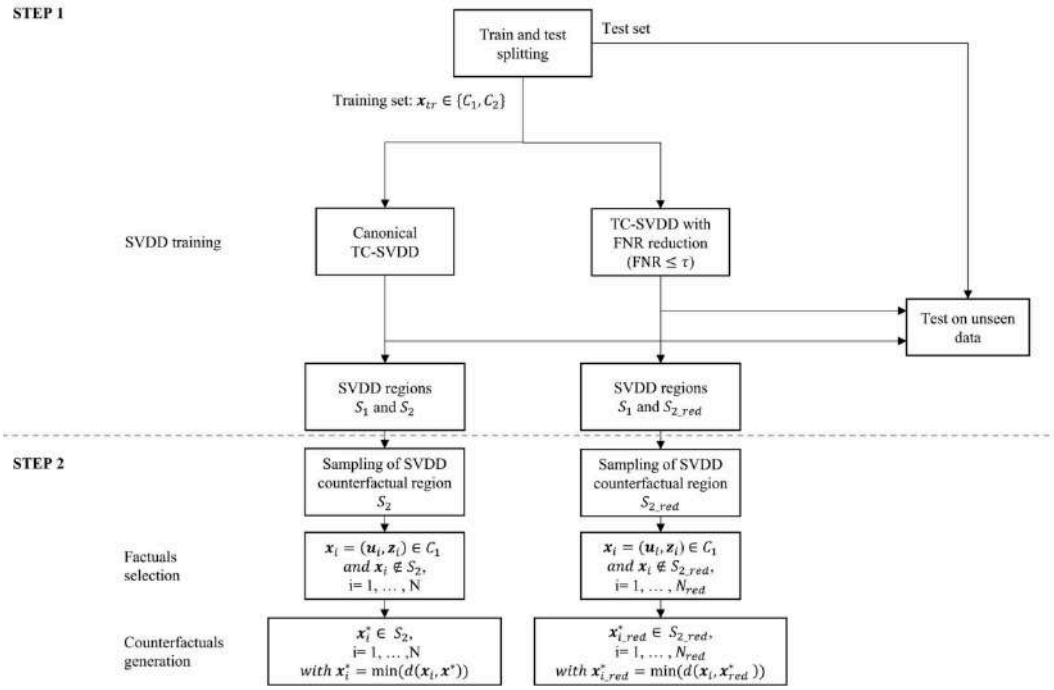


Figure 9.1: Schematic workflow of the counterfactuals generation process.

The *counterfactuals evaluation* section of the survey dealt with the assessment of some examples of *highT2DM* patients (factuals) and the corresponding target changes in biomarkers proposed by the algorithm in order to reduce the risk. Two examples of this kind of questions are shown in Table 9.1 where two factuals (F1 and F2) and their counterfactuals (C1 and C2) are reported.
F1 represents a 63-year-old female patient with hypertension, with FBS above the prediabetes threshold, and slightly elevated BMI (i.e., overweight class). LDL is near the desired range (i.e., optimal if LDL <2.6 mmol/L), HDL is acceptable (i.e., optimal if HDL >1.3 mmol/L in women), TG and Total Cholesterol are in the desired

range (i.e., TG <1.7 mmol/L and Total Cholesterol <5.18 mmol/L) according to general guidelines [1]. The algorithm proposes to lower the risk of developing T2DM by targeting the values in C1, namely by reducing FBS, BMI, sBP, TG and Total Cholesterol by keeping the LDL and TG levels almost constant. All the experts agreed that the proposed target values are reasonable to obtain a risk reduction when focusing on T2DM (i.e., 5 Moderately agree; 2 Strongly agree).
F2 represents a 55-year-old male patient living with hypertension, with FBS slightly above the prediabetes threshold and very high BMI (i.e., in the severe obesity range). LDL is near optimal, HDL is optimal (i.e., optimal if HDL >1.0 mmol/L in men), TG and Total Cholesterol are above the desired range. The algorithm proposes to lower the risk of developing T2DM by targeting the values in C2, namely by reducing FBS, BMI, sBP and TG while keeping the other values almost constant. In this case, experts expressed different opinions about the proposed risk reduction strategy (i.e., 3 Moderately disagree; 2 Moderately agree; 2 Strongly agree).

Table 9.1: **Counterfactuals evaluation.**

	Gender	**Age**	**FBS** $[mmol/L]$	**BMI** $[kg/m^2]$	**sBP** $[mmHg]$	**LDL** $[mmol/L]$	**HDL** $[mmol/L]$	**TG** $[mmol/L]$	**Total Chol** $[mmol/L]$	**HTN**
$F1: highT2DM$	Female	63	6.2	28.7	133	3.1	1.1	1.5	4.9	Yes
$C1: lowT2DM$			4.5	25	114	3.0	0.8	0.4	3.8	
$F2: highT2DM$	Male	55	5.8	44.1	157	3.0	1.2	2.3	5.9	Yes
$C2: lowT2DM$			5	40	134	3.0	1.2	2.0	6.2	

[1] https://www.mayoclinic.org/tests-procedures/cholesterol-test/about/pac-20384601

Chapter 10

CE for Intelligent Transportation Management in the Smart City

In this chapter of my thesis, I report another meaningful application of the counterfactual theory developed during my Ph.D.. In this case we applied counterfactual explanations to understand what are the subtle reasons that govern crowding in the metro subway of the city of Genoa. The study focused on the De Ferrari - Hitachi stop and it has been carrying on in the contest of MTT (More Than This) project (See Research Projects in the preface devoted to the **Publications**).

10.1 Introduction

Today, the cities we live in are far from being truly smart: overcrowding, pollution and poor transportation management are still in the headlines. With wide-scale deployment of advanced Artificial Intelligence (AI) solutions, however, it is possible to reverse course and apply the appropriate and necessary countermeasures to take a step forward on the road to sustainability. In this research, explainable AI techniques are applied to provide public transportation experts with suggestions on how to control crowding on subway platforms by leveraging interpretable rule-based models enhanced with counterfactual explanations. Numerical results for both the classification task and counterfactual properties encourage the goodness of the approach, but more importantly, an assessment of the quality of the proposed explainable methodology was submitted to a team of experts in the field to certify and validate the model. The experimental scenario relies on agent-based simulations of the De Ferrari Hitachi subway station of Genoa, Italy.

10.2 Methodology

The main objective of this work was to combine explainable-by-design and post-hoc XAI techniques for the short-term prediction of crowding conditions in specific subway areas (i.e., the platforms) using a dataset derived from simulations. To the best of my knowledge, this is the first work that combines rule-based interpretable models with counterfactual explanations to (i) predict possible crowding situations and (ii) suggest quantitative actions to prevent those situations based on what-if scenarios. This preliminary analysis will focus on a simple but straightforward use case in the city of Genoa, Italy.
The Genoa subway system (Figure 10.1) is a double-track single line of 7.1 km (4.4 mi) that connects the two main valleys of Genoa (Val Bisagno to the northeast with the Brignole stop and Valpolcevera to the northwest with the Brin stop) via the city center. The analysis will be devoted to the prediction of potential crowding situations in the De Ferrari Hitachi subway station, located below the main square of the city.
The dataset utilized contains simulations of the De Ferrari Hitachi subway station of Genoa, Italy was used. The dataset contains 28 variables (summarized in Tables 10.1,10.2 and 10.3) derived from 12696 simulations of 2 hours each. The simulations were generated using an agent-based model that allows to simulate the individual behavior of each single passenger and its interaction with other passengers and the surrounding environment based on parameters measured on-site or agreed upon interactions with stakeholders. In particular, the range of input parameters was set based on field-assessed values on weekdays, during off-peak hours. This simulation approach proved very useful in generating a sufficiently large set of realistic simulated scenarios in a cheaper and less time consuming way with respect to on-field experimental data collection. The dataset was generated within the framework of project More Than This[1].
The dataset was used to characterise the parameters related to a situation of potential crowding and suggest which values to act on (quantitatively) in the short run, to obtain the alternative uncrowded scenario i.e., its counterfactual.

Since we are interested in predicting the level of crowding on the two subway platforms of the subway station (i.e., towards Brin and towards Brignole) at time t (i.e., end of the simulation), we sampled the simulation data with a time interval Δt of 15 minutes by defining a time window of dimension 2, $[t-2\Delta t, t-\Delta t]$, i.e., considering the situation of the simulated subway station 30 and 15 minutes before the instant we would like to predict. Based on the simulated data, a *critical crowding threshold THR* of 30 people was selected and used as a discriminating value to identify the output of the classification problem. Having defined this threshold, 2 possible scenarios can thus be tested for each platform: average number of people waiting at the platform lower than *THR* (class 0) and average number of people waiting at the platform greater than *THR* (class 1). Based on the available data, the following

[1] `https://smarttrack.io/progetti/more-than-this/`

Figure 10.1: Genoa city subway system.[2].

distributions of output classes result:

- platform towards Brin: 6509 simulations belonging to class 0, 6187 simulations belonging to class 1.
- platform towards Brignole: 11718 simulations belonging to class 0, 978 simulations belonging to class 1.

As shown in Figure 10.1, De Ferrari Hitachi subway station is only one stop away from Brignole station, therefore, a smaller number of critical cases (i.e., points belonging to class 1) on the corresponding platform was considered plausible.
A subset of 7 variables was selected to be used in the counterfactual analysis and denoted as $V1, \ldots, V7$. Following interaction with transportation experts and feature ranking analysis, these variables were considered meaningful to ensure a trade-off between ability to represent the evolution of the crowding scenario and clarity of the graph. To avoid redundancy in the text, the subset of variables is listed in Table 10.1,10.2 and 10.3.

10.3 Counterfactual eXplanations

To make the counterfactual analysis even more specific, three different, alternative counterfactual explanations were generated for each input data, obtained by applying different constraint conditions to some of the input variables (i.e., imposing the no-variation condition to a subset of features, in the counterfactuals search algorithm):

- Unconstrained counterfactuals (C): are the counterfactual explanations obtained without imposing any constraint on the input data, i.e., allowing all features to vary.

[2]Attribution: This file is licensed under the Creative Commons Attribution-Share Alike 2.0 Generic license. Creator: Arbalete.

Table 10.1: Common features capturing the two time intervals of interest: minimum value, maximum value, mean, and standard deviation. $V1, \ldots, V7$, denotes the variables used in the counterfactual analisys.

	Name	Min	Max	Mean	Std	Description
Common	$\text{PIS}_{t-2\Delta t}$	36	711	347	176	**P**assenger **I**nflow from **S**tairs 30 minutes before, [passengers/¼h] (**V1**)
	$\text{PIS}_{t-\Delta t}$	37	713	345	178	**P**assenger **I**nflow from **S**tairs 15 minutes before, [passengers/¼h] (**V2**)
	$\text{PIE}_{t-2\Delta t}$	0	7	4	2	**P**assenger **I**nflow from **E**levator 30 minutes before, [passengers/¼h]
	$\text{PIE}_{t-\Delta t}$	0	8	3	2	**P**assenger **I**nflow from **E**levator 15 minutes before, [passengers/¼h]
	$\text{API}_{t-2\Delta t}$	3	79	38	14	**A**verage number of **P**assengers on the **I**ntermediate level 30 minutes before, [passenger]
	$\text{API}_{t-\Delta t}$	4	78	38	14	**A**verage number of **P**assengers on the **I**ntermediate level 15 minutes before, [passenger]
	$\text{MPI}_{t-2\Delta t}$	1	24	8	3	**M**aximum number of **P**assengers on the **I**ntermediate level 30 minutes before [passenger]
	$\text{MPI}_{t-\Delta t}$	1	24	8	3	**M**aximum number of **P**assengers on the **I**ntermediate level 15 minutes before, [passenger] (**V7**)

- Counterfactuals constrained on People-related features (C_{CP}): are the counterfactual explanations obtained by constraining the features more strictly related to people flow, namely $V1$, $V2$, and $V7$.
- Counterfactuals constrained on Trains-related features (C_{CT}): are the counterfactual explanations obtained by constraining the features related to trains, namely $V3$, $V4$, $V5$, and $V6$.

We remark that the subset of train-related features depends specifically on each model, i.e., the variables $V3, V4, V5$ and $V6$ refer to the same feature but specialized for the two platforms. As an example, in Section 10.5 the counterfactual explanations of two different simulated scenarios are shown, one for each travel direction (Brin, Brignole). To quantitatively evaluate the proposed counterfactual explanations in terms of their ability to be distinguished from data points in the factual class discriminative power was calculated, as defined in [163]. The general structure of the methodology is summarized in the flowchart in Figure 10.2. The system consists of a training phase in which data on the times Δt and $2\Delta t$ prior to the desired time for prediction are collected from the simulations, processed, and sent to the SVDD to be labeled. Indicating with $\mathcal{X}_t$ the training set at time t, we denoted by $f : \mathcal{X}_t \longrightarrow \{0, 1\}$ the function representing the SVDD prediction. THR is the critical threshold that triggers the classification, as explained above. Then, the operational phase acts on the vector of information at time $\tilde{t}$, $\boldsymbol{x}_{\tilde{t}}$, labeled in the

Table 10.2: Brignole features capturing the two time intervals of interest: minimum value, maximum value, mean, and standard deviation. $V1, \ldots, V7$, denotes the variables used in the counterfactual analisys.

	Name	Min	Max	Mean	Std	Description
Brignole	$BgTI_{t-2\Delta t}$	5	15	10	4	Average **T**ime **I**nterval between successive trains on the platform in Brignole direction 30 minutes before, [min] (**V3**)
	$BgTI_{t-\Delta t}$	5	15	10	4	Average **I**nterval between successive **T**rains on the platform in Brignole direction 15 minutes before, [min](**V4**)
	$BgPB_{t-2\Delta t}$	8	412	190	81	Average number of **P**assengers on the train **B**efore the stop on the platform in Brignole direction 30 minutes before, [passenger]
	$BgPB_{t-\Delta t}$	6	412	193	80	Average number of **P**assengers on the train **B**efore the stop on the platform in Brignole direction 15 minutes before, [passenger] (**V5**)
	$BgPGO_{t-2\Delta t}$	7	71	37	16	Average number of **P**assengers **G**etting **O**ff the train on the platform in Brignole direction 30 minutes before, [passenger]
	$BgPGO_{t-\Delta t}$	7	72	36	17	Average number of **P**assengers **G**etting **O**ff the train on the platform in Brignole direction 15 minutes before, [passenger]
	$BgTA_{t-2\Delta t}$	1	414	193	82	Average number of passengers on the **T**rain **A**fter departing from De Ferrari station in Brignole direction 30 minutes before, [passenger]
	$BgTA_{t-\Delta t}$	1	414	170	81	Average number of passengers on the **T**rain **A**fter departing from De Ferrari station in Brignole direction 15 minutes before, [passenger]
	$BgPP_{t-2\Delta t}$	0	95	12	11	Average number of **P**assengers waiting at the **P**latform in Brignole direction 30 minutes before, [passenger]
	$BgPP_{t-\Delta t}$	0	109	13	12	Average number of **P**assengers waiting at the **P**latform in Brignole direction 15 minutes before, [passenger] (**V6**)

training phase, creating the counterfactual example in the case when the subway is crowded. The actions of the counterfactual example will be visible in the subsequent time intervals, $\Delta\tilde{t}$ and $2\Delta\tilde{t}$ depending on the changed variables.

10.3.1 Application grounded evaluation

XAI methods have shown great potential in increasing user confidence in automatic decision models, however, how to evaluate those techniques is still a matter of debate. One of the most straightforward way is to perform an application grounded

Table 10.3: Brin features capturing the two time intervals of interest: minimum value, maximum value, mean, and standard deviation. $V1, \ldots, V7$, denotes the variables used in the counterfactual analisys.

	Name	Min	Max	Mean	Std	Description
Brin	$\mathrm{BrTI}_{t-2\Delta t}$	5	15	10	4	Average **T**ime **I**nterval between successive trains on the platform in Brin direction 30 minutes before, [min] (**V3**)
	$\mathrm{BrTI}_{t-\Delta t}$	5	15	10	4	Average **T**ime **I**nterval between successive trains on the platform in Brin direction 15 minutes before, [min] (**V4**)
	$\mathrm{BrPB}_{t-2\Delta t}$	1	412	180	87	Average number of **P**assengers on the train **B**efore the stop on the platform in Brin direction 30 minutes before, [passenger]
	$\mathrm{BrPB}_{t-\Delta t}$	1	412	180	87	Average number of **P**assengers on the train **B**efore the stop on the platform in Brin direction 15 minutes before, [passenger] (**V5**)
	$\mathrm{BrPGO}_{t-2\Delta t}$	0	16	8	17	Average number of **P**assengers **G**etting **O**ff the train on the platform in Brin direction 30 minutes before, [passenger]
	$\mathrm{BrPGO}_{t-\Delta t}$	0	17	7	17	Average number of **P**assengers **G**etting **O**ff the train on the platform in Brin direction 15 minutes before, [passenger]
	$\mathrm{BrTA}_{t-2\Delta t}$	3	415	209	87	Average number of passengers on the **T**rain **A**fter departing from De Ferrari station in Brin direction 30 minutes before, [passenger]
	$\mathrm{BrTA}_{t-\Delta t}$	3	414	211	88	Average number of passengers on the **T**rain **A**fter departing from De Ferrari station in Brin direction 15 minutes before, [passenger]
	$\mathrm{BrPP}_{t-2\Delta t}$	0	213	36	26	Average number of **P**assengers waiting at the **P**latform in Brin direction 30 minutes before, [passenger]
	$\mathrm{BrPP}_{t-\Delta t}$	0	216	37	26	Average number of **P**assengers waiting at the **P**latform in Brin direction 15 minutes before, [passenger] (**V6**)

evaluation, that is, to assess the quality of explanations in their applicative context, involving domain experts. A team of 5 experts in the field of transportation and logistics that possess only basic AI knowledge was asked to fill out a Microsoft Forms survey anonymously. Participation was completely voluntary. First, the experts were asked to evaluate four scenarios showing the average values of variables $V1$-$V7$ for each specific output class and each specific model. The experts were blinded to the actual output class and were asked to select whether each scenario corresponded to a situation with a number of people on the platform below or above *THR*. They were also asked to specify their level of confidence on a 4-level scale.

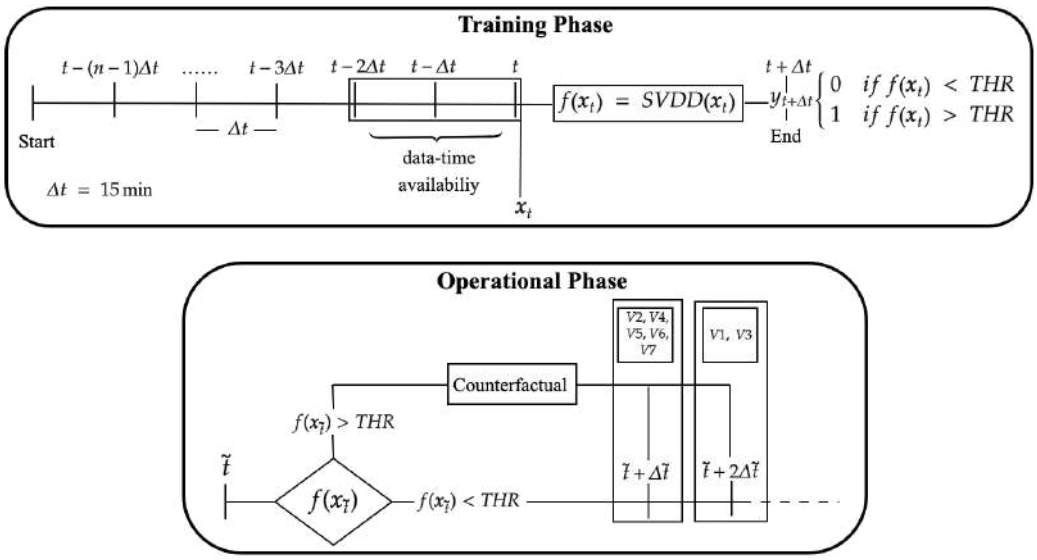


Figure 10.2: Methodology flow chart.

This first part of the questionnaire aimed to assess whether the chosen features and the output were considered sufficiently explanatory of the problem to be modeled. Then, the experts were asked to evaluate four examples of factuals with the corresponding counterfactuals C, C_{CP} and C_{CT} (2 related to Brin station and 2 related to Brignole station). For each example the experts were asked to specify the level of agreement with the proposed suggestions on a scale of 1 to 5 and to provide a brief justification (non-mandatory field). In addition, the expert had to specify which of the 3 proposed solutions was considered the best. Finally, each expert was asked to assess the realism and applicability of the results and to provide overall feedback on the proposed methodology. In addition, experts were asked to evaluate which features, among those considered in the model, are most easily controllable in the short run. In this regard, they were also asked to suggest any additional variables to be considered in a possible follow-up of the study.

10.4 Results

10.4.1 LLM for crowding prediction

Two separate LLMs (one per platform), were trained on 70% of the data and tested on the remaining 30%. Accordingly, we will refer to two distinct models: LLM_{Bg} aims to predict the state of crowding on the platform in the Brignole direction, whereas LLM_{Br} focuses on predicting crowding on the platform in the Brin direction. The classification performance via LLM (for each model) are reported in Table 10.4 Table 10.5 reports the main characteristics of LLM_{Bg} and LLM_{Br} in terms of number of decision rules, covering and error.
$\text{BgPP}_{t-2\Delta t}$, $\text{PIS}_{t-2\Delta t}$, and $\text{BgTI}_{t-\Delta t}$ were particularly decisive in predicting the ex-

Table 10.4: Performance results of LLM_{Bg} and LLM_{Br}.

Model	training accuracy	test accuracy	sensitivity (on test set)	specificity (on test set)
LLM_{Bg}	0.82	0.82	0.73	0.83
LLM_{Br}	0.75	0.70	0.71	0.69

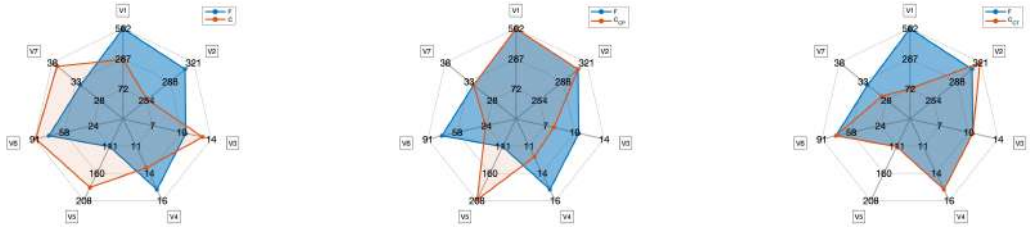
ceedance of *THR* in the LLM_{Bg} model (i.e., feature ranking > 0.2), whereas the most relevant variables for the LLM_{Br} model are $\text{PIS}_{t-2\Delta t}$, $\text{PIE}_{t-\Delta t}$, $\text{PIS}_{t-\Delta t}$, and $\text{API}_{t-\Delta t}$, all of which are variables closely related to the flow of passengers entering and circulating through the station in the 2 previous intervals. Feature ranking demonstrates once again how the number of passenger at the platform is largely influenced by the flow of passengers entering and stationing within the subway station in the 2 time intervals considered (i.e., 15 or 30 minutes before the prediction instant), as well as by the trains frequency. The use of XAI techniques such as the LLM allows for a more in-depth exploration of these intuitive considerations, by providing quantitative thresholds in the form of a value ranking. For example, the value ranking provides thresholds equal to 27 for $\text{BgPP}_{t-2\Delta t}$, 538 for $\text{PIS}_{t-2\Delta t}$, and 14 for $\text{BgTI}_{t-\Delta t}$ when applied to the LLM_{Bg} model. This means that in general, values of these variables above the identified thresholds are associated with a higher probability of providing an output of 1 in the model and therefore associated with a situation of potential crowding.

Table 10.5: Main characteristics of LLM_{Bg} and LLM_{Br}: # of rules, covering and error.

Model	# of rules	$C(R_i)$ (mean ±s.d.)	$E(R_i)$ (mean ±s.d.)
LLM_{Bg}	34 (23;11)	11.00%± 8.14%	4.6%± 0.51%
LLM_{Br}	50(25;25)	7.20%± 4.46%	4.77%± 0.65%

10.4.2 Evaluation of counterfactual explanations

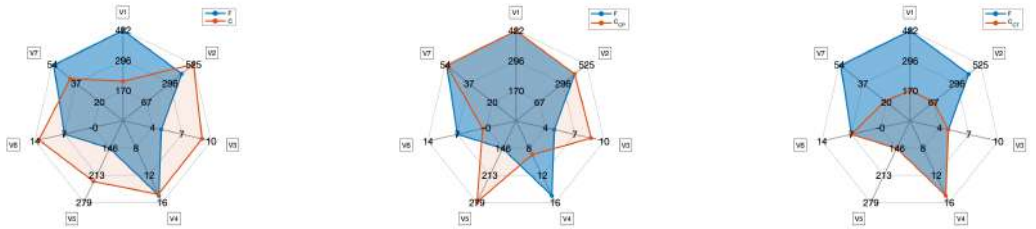
A set of factuals was extracted from test records having output equal to 1 (i.e., 1051 for the Brin travel direction and 214 for the Brignole travel direction) and counterfactual explanations for each of the three typologies described were generated for each factual. The discriminative power of counterfactual explanations generated for the Brin travel direction was of about 90.6%, 91.8%, and 93.9% for C, C_{CP} and C_{CT}, respectively. The discriminative power of counterfactual explanations generated for the Brignole travel direction was on average slightly lower compared to that of Brin (86.7%, 94.5%, and 89.8% for C, C_{CP} and C_{CT}, respectively).



(a) **Brin** - unconstrained (b) **Brin** - constraints on people (c) **Brin** - constraints on trains

Figure 10.3: Spiderplot of the three proposed scenarios containing the factuals and respective counterfactuals for the platform in Brin direction.

The average survey completion time was of about 18 minutes. Despite reporting minimal or basic knowledge in AI, respondents believe that AI will play a pivotal role in crowd management in public environments. In the first series of four questions the experts were asked to select the output class given a set of 7 features ($V1$–$V7$) describing a specific scenario characterized by an output equal to 0 or an output equal to 1 in a specific platform, as shown in Table 10.6. All experts correctly assessed case A as a situation where the number of people on the platform is above the threshold (output = 1). Experts said they had a fairly high (3 out of 5) or high (2 out of 5) confidence in the answer given. Similarly, almost all the experts (i.e., 4 out of 5) correctly assessed case D as belonging to output class 1, although with a decrease in reported confidence (low confidence: 2; fairly high confidence: 2; high confidence: 1). Case B was correctly classified by 3 out of 5 experts as belonging to output class 0 (low confidence: 2; fairly high confidence: 3). Finally, case C was correctly classified only by 2 out of 5 experts as belonging to output class 0 (low confidence: 1; fairly high confidence: 4). In 3 out of 4 examples, experts were able to correctly predict the output class; the output class 1 was predicted more accurately,



(a) **Brignole** - unconstrained (b) **Brignole** - constraints on people (c) **Brignole** - constraints on trains

Figure 10.4: Spiderplot of the three proposed scenarios containing the factuals and respective counterfactuals for the platform in Brignole direction.

Table 10.6: Average values of variables $V1$-$V7$ on the training set, for each specific output class and each specific model.

	Brin		Brignole	
Feature	A	B	C	D
$V1$	418	280	342	477
$V2$	386	316	345	386
$V3$	10	10	10	10
$V4$	11	9	10	14
$V5$	218	201	173	174
$V6$	42	32	11	22
$V7$	38	38	38	38
Output	1	0	0	1

by an higher number of experts, although experts were rarely completely confident in the answer given. Then, the experts were asked to evaluate a set of counterfactual explanations. One example for each platform is reported in Figure 10.3 and Figure 10.4, respectively. As for the example shown in Figure 10.3, concerning the platform in Brin direction, the majority of experts were found to agree with the proposed suggestions (moderately agree: 3; neither agree nor disagree: 1; moderately disagree: 1). C_{CP} was judged by experts to be the most realistic solution, as it suggests preventing a possible crowded situation on the platform by reducing $V3$ and $V4$ by 3 minutes, that is, reducing the interval between trains in the previous two time windows. Furthermore, the presence of fewer people on the platform at time $t-1$ (lower $V6$) is associated with a lower probability of crowding at time t. In contrast, the suggestion proposed by counterfactual C was not considered realistic since the passengers inflow in the previous two time intervals ($V1$ and $V2$) is reduced, but at the same time there is a counter intuitive increase in the number of people waiting at the platform ($V6$).

As we can observe by comparing Figure 10.3 and Figure 10.4 the suggested trends of variation are mostly comparable in the two platforms, however, there are some exceptions worth noting. For example, in these specific scenarios, the variables $V2$ and $V7$ have opposite behavior when considering C in the two platforms: this might suggest that the crowding condition is more related to the combination of passengers on the stairs and at the platform rather than the number of passengers in a specific station area. Focusing on the example shown in Figure 10.4, 4 out of 5 experts moderately agreed with the proposed suggestion, whereas only one expert was neutral. Also in this example C_{CP} was considered the most realistic solution. The

countermeasures considered most effective in achieving the values suggested by the counterfactual explanations include turnstiles blockage for reducing station access and a reorganization of the timetable to time intervals between consecutive trains. In general, counterfactual explanations were considered realistic by all the experts, however they were not always considered readily applicable (realistic and applicable: 3; realistic but not applicable: 2). Among the variables considered in the simplified simulation scenario, the passengers inflow was considered the most controllable variable in the short-run (15-30 minutes) (4 votes), followed by the number of people boarding the train and train frequency (2 votes each). Additional controllable variables suggested by the surveyed experts include the waiting time at the platform, the number of carriages per train and the train length of stay at the station.

10.5 Final Comments

10.5.1 LLM for crowding prediction

In this work, LLM has shown the ability to predict the evolution of crowding in a given station area (i.e., a specific subway platform, in this case) by having information on the incoming, outgoing, and current passenger flow of the platforms in a previous time window. Prediction accuracy can be considered satisfactory, with values above 80% when considering LLM_{Bg} and slightly lower values (around 70%) when considering LLM_{Br}. The two models are characterized by a quite high number of rules that can sufficiently represent both classes, with a covering that can reach up to 30% and an error associated with individual rules lower than 5%. Rule-based models can be further refined by filtering out redundant rules or conditions and merging similar rules, allowing the logic underlying knowledge extraction to be streamlined while maintaining satisfactory predictive performance.
Rule-based approaches have been already used in the context of passenger flow prediction. For example, Zhao et al. [131] explored the influence of temporal, spatial and external features in predicting passenger flow within a day using tree based ensemble methods (random forest and gradient boosting decision tree) on data from the Shanghai Metro Automatic Fare Collection (AFC) system. However, feature ranking was used only for feature selection purposes. In our work, the analysis of feature ranking similarly allowed to identify the main features that the model uses to predict a particular output. In addition, the further value ranking analysis allowed to quantitatively specify the values of those features that are most determinant for a certain output. In particular, the value ranking given for LLM_{Bg} and reported as an example in Section 10.4.1 are similar or slightly higher with respect to the average values for output equal to 1 (e.g., V2 and V4 in case D, Table 10.6) but definitely higher with respect to the average values for output equal to 0 (e.g., V2 and V4 in case C, Table 10.6), thus showing high discrimination capabilities between the two classes. This analysis has enabled the identification of global discrimination thresholds related to individual features, however, the end user could benefit from an

additional analysis of individual scenarios through local explanations. An extremely useful tool is therefore the generation of counterfactual explanations that provide quantitative suggestions by varying multiple features simultaneously while focusing on a single scenario of interest.

10.5.2 Counterfactual explanations for crowding prevention

The quality of the set of counterfactual explanations was verified both quantitatively through the calculation of discriminative power and qualitatively by consulting expert opinion by means of a questionnaire. The discriminative power is around 90% for C, C_{CP} and C_{CT} in both platforms, hence, the set of explanations belonging to class 0 can be accurately distinguished from the source class of factuals (class 1). Discriminative power allows explanations to be validated from a computational point of view, however, to verify the actual applicability of the method this metric was not sufficient and interaction with experts was necessary. According to the experts, the suggestions produced through counterfactual explanations can be considered as realistic, however, in the future it might be useful to consider additional controllable features, such as train dwell time at the station and the number of carriages per train which could possibly be added if the station is expected to be significantly crowded. An additional interesting insight that emerged from the questionnaire is that the suggested changes may not systematically be applicable in the short run, as the logistic infrastructure may not be able to intervene quickly enough (e.g., increase train capacity, dynamically control station access). This aspect was in part considered through the introduction of different explanations focusing on different subgroups of features and can be further developed through iterative interaction with the stakeholders.

10.5.3 Limitations and future research

In this study, the method was applied to a specific station location, but it can be easily generalized to other areas of the station such as entrances, and emergency exits. Moreover, in this preliminary study, a fairly low critical crowding threshold (30 people on the platform) was chosen based on considerations due to the chosen facility and its normal passenger flow. In fact, the objective of the study is to predict potential crowding in everyday situations, in the short term, whereas the presence of exceptional events with excessively higher than normal flows (e.g., events, concerts, soccer games) is known with due advantage and managed differently. However, it is important to note that the proposed analysis may be easily applied to different threshold values. Future developments of the study may cover different aspects, such as the extension of the prediction window to consider possible inner dynamics in the medium to long term, the comparison of counterfactual explanations obtained with different critical crowding threshold levels or the customization of the set of controllable and non-controllable features defined based on requirements defined together with the transportation infrastructure stakeholders. Furthermore, expert

comments highlighted the need to analyze the causal relationships between variables in order to obtain more realistic suggestions.

Part IV

Conclusion and Future Works

Chapter 11

Conclusions and Future Works

This chapter briefly summarizes the contributions of this dissertation and presents some future lines of work in each field.

- **Part I** is the core of my PhD, where I gave the major contributions to the body of knowledge. The ideas underneath the studies I developed in this field is i) to give the possibility to control the behavior of a ML algorithm on the basis of the input features (safety regions) ii) to provide ML algorithms I developed with probabilistic guarantees iii) to make all this framework explainable.
  Specifically, the contributions of each chapter are:
  - **Chapter 2** introduces the concept of safety region, intended as a subspace of the input space where guarantees on the output can be provided. Moreover, in this work I developed a clear and operational framework based on SVDD that has been the basis also for other novelties in my research.
  - Research presented in **Chapter 3**, **Chapter 4** and **Chapter 5** exploit the robust background of Conformal Prediction and Probabilistic Scaling to make a step further in i) application of online guarantees to classification algorithms ii) understanding the relationship between exponential distributions and safety regions and iii) providing conformity framework for explainable AI. What it is worth to underline is the development of the concept of *scalable classifier* that establishes clearly a new family of classification algorithms sharing the property of being controllable by a scalable parameter.
- The results discussed in **Part II** constitute as significant a part of my research as those in Part I. The way in which I delved into counterfactual explanations starts exactly from the idea of safety region, i.e. controlling a ML algorithm such that the output performs what the machine learner desires. In this perspective, I presented two methodologies to retrieve counterfactual explanations from tabular data: binary CEs (**Chapter 6**) and multi-class CEs (**Chapter 7**). The greatest novelties borrowed by both the methods are i) SVDD allows

to define closed envelops for the research of CEs ii) an analytical solution for CEs can be found (at least in the linear case) iii) quasi-random sampling is less expensive but equally functional than grid-search for numerical approximation of CEs. Specifically for Chapter 7, multi-counterfactual explanations modeling has been proved to be a good generalization of the binary case, showing that the good properties defined in the binary framework can be easily scaled in the multi-class case. Moreover all the concepts developed in this research dealt with questions and open topics in CE, for example the already cited definition of "minimal change". My answer is simply the solution of a minimization problem properly defined.
Challenges and hot topics to investigate are however ahead:

  - definition of agnostic models CEs: the quasi-random sampling allows to retrieve CEs from any classifier, so it is worth to test classifiers different from SVDD and understand how CEs change;
  - incremental CEs: in the multi-class case, where it is possible to define a sequence of states for the output (for example the severity of a disease), it would be worth understanding how the same factual gradually changes from label to label (i.e., from condition to condition), defining a sequence of counterfactual;
  - assessing the conformity of CEs through conformal prediction.

- Finally, **Part III** reports the main applications of the methodologies developed during the PhD. All the works presented in this part relate specifically to one of the algorithms or frameworks outlined in Part I and Part II. In detail, considering each work presented by chapter

  - **Chapter 8** shows how SafeSVDD can be used to detect adversarial machine learning attacks and make countermeasures to prevent them. The results validated on three different attacks (Carlini-Wagner, Jacobi based Saliency Map and Fast Gradient Sign Method) prove that the safe framework provided by SVDD and its controllable variations are a valid approach to detect and contrast adversarial attacks. Another important contribution in this work has been the possibility to extract intelligible rules from the classification, making all the methodology totally explainable. However there are some open questions and challenges that naturally arise from this work, for example
    * application of the safe framework to adversarial attacks on images;
    * online detection of attacks and not "a posteriori" like in the current framework;
    * scalability to large dataset.
  - **Chapter 9** and **Chapter 10** both report the application of counterfactual explanations, respectively, in a health care domain (prevention

of type 2 diabetes) and in an intelligent transportation system domain (prediction of crowded situations in the subway). Specifically

  * it has been proved (and medically validated by real doctors through questionnaire) that CEs can provide valid and meaningful recommendations to prevent the insurgence of the disease.
  * CEs can be extracted and validated to make online corrections to the management to crowd flow management in subways.

In summary, the research I have done during my Ph.D. has focused mainly on the need for AI systems to have controllability and explainability. The work done so far has helped me understand how tough this task can be. In order to give an accurate and reliable methodology that can provide such a framework, several aspects and disciplines need to be considered: from probability and design control to ethics and data analysis. With my work, which has covered all these fields, I hope to have made my contribution to the body of knowledge on these topics. Anyway, it has been fundamental to my formation.

Remark. During my Ph.D. I also delved into the fields of Computer Vision, Deep Learning, and Cyber-Physical Systems for imaging. Since it is not closely related to the topic covered in this thesis, I have decided to not talk about that. However, they have been an important part of my PhD and personal training, both academically and industrially. All activities related to this topic were carried out together with Aitek, which took care of my industrial training.
To summarize, very briefly, the activities and research in this field, I report

- the development of a deep learning architecture for the detection of artichoke plants through drone: we built a single shot detector composed by a convolutional backbone enhanced by a Feature Pyramid Network to extract information features at different levels of high (due the use of the drone). The obtained results, validated on ground, proved the good efficiency of the network as reported in the publication [177]. Moreover post-processing techniques like temporal-tracking have been used to improve the detection and the classification of the network.

- As mentioned in the introduction of this thesis, I have worked on several projects involving cyber-physical systems and machine learning solutions. The activities carried out in these projects have been fundamental to my scientific education, including the project management part. In this regard, I have participated in several European electronics forums, such as the Key Digital Technology (KDT) or the European forum for electronic components and systems, sharing ideas and discussing new developments in electronics.

The **future lines** I would like to follow only partially cover the works addressed so far. Implementations and variations of counterfactual explanations or conformal predictions (for example how to implement/solve the difficult optimization problems presented in the theory), from my point of view the major contributions of my

Ph.D., need to be investigated but cannot be the only guidelines of my future academic works. In this sense, I am moving into the study of Physics Informed Neural Networks, a rather innovative methodology that fuses together physics-based systems and data-driven approaches. In particular, the problem I am trying to address is how to simplify complex systems of differential equations using data, without losing precision in the solution. This is the main topic of my study abroad at the University of California Berkeley where I am right now writing this thesis and that I'd like to carry on.

Bibliography

[1] J. Mercer. "Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 209 (1909), pp. 415–446. ISSN: 02643952. URL: http://www.jstor.org/stable/91043 (visited on 04/26/2022).

[2] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[3] Jacob Cohen. *A Coefficient of Agreement for Nominal Scales*. 1960. DOI: 10.1177/001316446002000104. URL: http://dx.doi.org/10.1177/001316446002000104.

[4] Manuel Blum et al. "Time Bounds for Selection". In: *J. Comput. Syst. Sci.* 7.4 (Aug. 1973), pp. 448–461. ISSN: 0022-0000. DOI: 10.1016/S0022-0000(73)80033-9. URL: https://doi.org/10.1016/S0022-0000(73)80033-9.

[5] Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks". In: *Mach. Learn.* 20.3 (Sept. 1995), pp. 273–297. ISSN: 0885-6125. DOI: 10.1023/A:1022627411411. URL: https://doi.org/10.1023/A:1022627411411.

[6] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[7] Shigeo Abe. "Support Vector Machines for Pattern Classification". In: *Advances in Pattern Recognition*. 1999.

[8] D.P. Bertsekas. *Nonlinear Programming*. 2nd ed., Athena Scientific, Belmont, MA, 1999.

[9] I.A. Taha and J. Ghosh. "Symbolic interpretation of artificial neural networks". In: *IEEE Transactions on Knowledge and Data Engineering* 11.3 (1999), pp. 448–463. DOI: 10.1109/69.774103.

[10] D. Tax and R. Duin. "Support vector domain description". In: *Pattern Recognition Letters 20* (1999), pp. 1191–1199.

[11] David M. J. Tax and Robert P. W. Duin. "Support vector domain description". In: *Pattern Recognition Letters* 20 (1999), pp. 1191–1199.

[12] V.N. Vapnik. "An overview of statistical learning theory". In: *IEEE Transactions on Neural Networks* 10.5 (1999), pp. 988–999. DOI: 10.1109/72.788640.

[13] E. Boros et al. "An implementation of logical analysis of data". In: *IEEE Transactions on Knowledge and Data Engineering* 12.2 (2000), pp. 292–306. DOI: 10.1109/69.842268.

[14] Bernhard Schölkopf. "The Kernel Trick for Distances". In: *Proceedings of the 13th International Conference on Neural Information Processing Systems*. NIPS'00. Denver, CO: MIT Press, 2000, pp. 283–289. DOI: 10.5555/3008751.3008793.

[15] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer: New York, 2000.

[16] Dmitri Burago, Yu. D. Burago, and Sergei O. Ivanov. "A Course in Metric Geometry". In: 2001.

[17] Cristiano Cervellera and Marco Muselli. "Deterministic Design for Neural Network Learning: An Approach Based on Discrepancy". In: *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* 15 (June 2004), pp. 533–44. DOI: 10.1109/TNN.2004.824413.

[18] L. Vandenberghe S. Boyd. *Convex Optimization*. Cambridge University Press, New York, 2004.

[19] D. Tax and R. Duin. "Support vector domain description". In: *Machine Learning* (2004), pp. 45–66.

[20] David M. J. Tax and Robert P. W. Duin. "Support Vector Data Description". In: *Machine Learning* 54 (2004), pp. 45–66.

[21] Roger Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005. DOI: 10.1017/CBO9780511754098.

[22] KiYoung Lee et al. "Improving support vector data description using local density degree". In: *Pattern Recognition* 38.10 (2005), pp. 1768–1771. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2005.03.020. URL: https://www.sciencedirect.com/science/article/pii/S0031320305001469.

[23] Marco Muselli. *Switching Neural Networks: A New Connectionist Model for Classification*. Jan. 2005. DOI: 10.1007/11731177_4.

[24] Marco Muselli and Alfonso Quarati. "Reconstructing positive Boolean functions with shadow clustering". In: vol. 3. Jan. 2005, III/377–III/380 vol. 3. ISBN: 0-7803-9066-0. DOI: 10.1109/ECCTD.2005.1523139.

[25] Marco Muselli. "Switching Neural Networks: A New Connectionist Model for Classification". In: *Neural Nets*. Ed. by Bruno Apolloni et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 23–30. ISBN: 978-3-540-33184-1.

[26] Haydemar Nunez, Cecilio Angulo, and Andreu Catala. "Rule-based learning systems for support vector machines". In: *Neural Processing Letters* 24.1 (2006), pp. 1–18.

[27] Syamal Sen, Tathagata Samanta, and Andrea Reese. "Quasi-versus pseudo-random generators: Discrepancy, complexity and integration-error based comparison". In: *Int J Innov Comput Info Control* 2 (Jan. 2006).

[28] Olivier Chapelle. "Training a Support Vector Machine in the Primal". In: *Neural Computation* 19.5 (2007), pp. 1155–1178. DOI: 10.1162/neco.2007.19.5.1155.

[29] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. "Kernel methods in machine learning". In: *The Annals of Statistics* 36.3 (June 2008). DOI: 10.1214/009053607000000677. URL: https://doi.org/10.1214%2F009053607000000677.

[30] Abhinav Saxena et al. "Damage propagation modeling for aircraft engine run-to-failure simulation". In: *2008 International Conference on Prognostics and Health Management*. 2008, pp. 1–9. DOI: 10.1109/PHM.2008.4711414.

[31] Abhinav Saxena et al. "Damage propagation modeling for aircraft engine run-to-failure simulation". In: *2008 International Conference on Prognostics and Health Management*. 2008, pp. 1–9. DOI: 10.1109/PHM.2008.4711414.

[32] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009. URL: http://www-stat.stanford.edu/~tibs/ElemStatLearn/.

[33] S. Abe. "Support Vector Machines for Pattern Classification". In: *Advances in Pattern Recognition* (2010).

[34] Nahla Barakat and Andrew P. Bradley. "Rule extraction from support vector machines: A review". In: *Neurocomputing* 74.1 (2010). Artificial Brains, pp. 178–190. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2010.02.016. URL: https://www.sciencedirect.com/science/article/pii/S0925231210001591.

[35] S. Fortunato. "Community detection in graphs". In: *Phys. Rep.-Rev. Sec. Phys. Lett.* 486 (2010), pp. 75–174.

[36] Guang-Xin Huang et al. "Two-class support vector data description". In: *Pattern Recognit.* 44 (2011), pp. 320–329.

[37] Guangxin Huang et al. "Two-class support vector data description". In: *Pattern Recognition* 44.2 (2011), pp. 320–329. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2010.08.025. URL: https://www.sciencedirect.com/science/article/pii/S0031320310004115.

[38] Marco Muselli and Enrico Ferrari. *Coupling Logical Analysis of Data and Shadow Clustering for Partially Defined Positive Boolean Function Reconstruction*. Jan. 2011. DOI: 10.1109/tkde.2009.206. URL: http://dx.doi.org/10.1109/tkde.2009.206.

[39] Marco Muselli and Enrico Ferrari. *Coupling Logical Analysis of Data and Shadow Clustering for Partially Defined Positive Boolean Function Reconstruction*. Jan. 2011. DOI: 10.1109/tkde.2009.206. URL: http://dx.doi.org/10.1109/tkde.2009.206.

[40] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[41] Jinghong Fang et al. "A SVDD method based on maximum distance between two centers of spheres". In: *Chinese Journal of Electronics* 21.1 (2012), pp. 107–111. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84863014315&partnerID=40&md5=f8b3ea5d64744b9e6030cc4db4f5b119.

[42] Davide Cangelosi et al. "Logic Learning Machine creates explicit and stable rules stratifying neuroblastoma patients". In: *BMC Bioinform.* 14.S-7 (2013), S12. DOI: 10.1186/1471-2105-14-S7-S12. URL: https://doi.org/10.1186/1471-2105-14-S7-S12.

[43] Cristiano Cervellera et al. "Quasi-random sampling for approximate dynamic programming". In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*. 2013, pp. 1–8. DOI: 10.1109/IJCNN.2013.6707065.

[44] Andreas Theissler and Ian Dear. "Autonomously Determining the Parameters for SVDD with RBF Kernel from a One-Class Training Set". In: *International Journal of Computer and Information Engineering* 7.7 (2013), pp. 949–957. ISSN: eISSN: 1307-6892. URL: https://publications.waset.org/vol/79.

[45] Guocheng Xie, Yun Jiang, and Na Chen. "A Multi-class Support Vector Data Description Approach for Classification of Medical Image". In: *2013 Ninth International Conference on Computational Intelligence and Security*. 2013, pp. 115–119. DOI: 10.1109/CIS.2013.31.

[46] V. N. Balasubramanian, S.S. Ho, and V. Vovk. *Conformal Prediction for Reliable Machine Learning*. 1st ed. 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann Elsevier, 2014. ISBN: 9780123985378.

[47] Ross Girshick et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. arXiv: 1311.2524 [cs.CV].

[48] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[49] Shehroz S. Khan and Michael G. Madden. "One-class classification: taxonomy of study and review of techniques". In: *The Knowledge Engineering Review* 29.3 (2014), pp. 345–374. DOI: 10.1017/s026988891300043x. URL: https://doi.org/10.1017%5C%2Fs026988891300043x.

[50] Lijian Xu et al. "Communication Information Structures and Contents for Enhanced Safety of Highway Vehicle Platoons". In: *IEEE Transactions on Vehicular Technology* 63.9 (2014), pp. 4206–4220. DOI: 10.1109/TVT.2014.2311384.

[51] Abdiansah Abdiansah and Retantyo Wardoyo. "Article: Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM". In: *International Journal of Computer Applications* 128.3 (Oct. 2015). Published by Foundation of Computer Science (FCS), NY, USA, pp. 28–34.

[52] M. Aiello, M. Mongelli, and G. Papaleo. "DNS tunneling detection through statistical fingerprints of protocol messages and machine learning". In: *International Journal of Communication Systems* 28.14 (2015), pp. 1987–2002. DOI: https://doi.org/10.1002/dac.2836. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/dac.2836. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.2836.

[53] M. Aiello, M. Mongelli, and G. Papaleo. "DNS tunneling detection through statistical fingerprints of protocol messages and machine learning". In: *International Journal of Communication Systems* 28.14 (2015), pp. 1987–2002. DOI: https://doi.org/10.1002/dac.2836. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/dac.2836. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.2836.

[54] Maurizio Aiello, Maurizio Mongelli, and Gianluca Papaleo. "DNS tunneling detection through statistical fingerprints of protocol messages and machine learning". In: *International Journal of Communication Systems* 28.14 (2015), pp. 1987–2002.

[55] Ross B. Girshick. "Fast R-CNN". In: *CoRR* abs/1504.08083 (2015). arXiv: 1504.08083. URL: http://arxiv.org/abs/1504.08083.

[56] Wei Liu et al. "SSD: Single Shot MultiBox Detector". In: *CoRR* abs/1512.02325 (2015). arXiv: 1512.02325. URL: http://arxiv.org/abs/1512.02325.

[57] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[58] Steven E Shladover et al. "Cooperative adaptive cruise control: Definitions and operating concepts". In: *Transportation Research Record* 2489.1 (2015), pp. 145–152.

[59] Yair Wiener and Ran El-Yaniv. "Agnostic pointwise-competitive selective classification". In: *Journal of Artificial Intelligence Research* 52 (2015), pp. 171–201.

[60] Songfeng Zheng. "Smoothly approximated support vector domain description". In: *Pattern Recognition* (2015).

[61] Lixiang Duan et al. “A new support vector data description method for machinery fault diagnosis with unbalanced datasets”. In: *Expert Systems with Applications* 64 (2016), pp. 239–246. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2016.07.039. URL: https://www.sciencedirect.com/science/article/pii/S0957417416303864.

[62] Dongyao Jia et al. “A Survey on Platoon-Based Vehicular Cyber-Physical Systems”. In: *IEEE Communications Surveys & Tutorials* 18.1 (2016), pp. 263–284. DOI: 10.1109/COMST.2015.2410831.

[63] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. “Deepfool: a simple and accurate method to fool deep neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2574–2582.

[64] Nicolas Papernot et al. “Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks”. In: May 2016, pp. 582–597.

[65] Nicolas Papernot et al. “The limitations of deep learning in adversarial settings”. In: *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE. 2016, pp. 372–387.

[66] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 2016, pp. 1135–1144.

[67] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “"Why Should I Trust You?": Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1135–1144. ISBN: 9781450342322. DOI: 10.1145/2939672.2939778. URL: https://doi.org/10.1145/2939672.2939778.

[68] Nicholas Carlini and David Wagner. “Towards evaluating the robustness of neural networks”. In: *2017 ieee symposium on security and privacy (sp)*. IEEE. 2017, pp. 39–57.

[69] Luca Caviglione et al. “Measuring the energy consumption of cyber security”. In: *IEEE Communications Magazine* 55.7 (2017), pp. 58–63.

[70] Scott M. Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4768–4777. ISBN: 9781510860964.

[71] Myeongsuk Pak and Sanghoon Kim. “A review of deep learning in image recognition”. In: *2017 4th international conference on computer applications and information processing technology (CAIPT)*. IEEE. 2017, pp. 1–3.

[72] Stefania Santini et al. "A Consensus-Based Approach for Platooning with Intervehicular Communications and Its Validation in Realistic Scenarios". In: *IEEE Transactions on Vehicular Technology* 66.3 (2017), pp. 1985–1999. DOI: 10.1109/TVT.2016.2585018.

[73] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. "Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR". In: *CoRR* abs/1711.00399 (2017). arXiv: 1711.00399. URL: http://arxiv.org/abs/1711.00399.

[74] Sandra Wachter, Brent Daniel Mittelstadt, and Chris Russell. "Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR". In: *Cybersecurity* (2017).

[75] T. Alamo, J. M. Manzano, and E. F. Camacho. "Robust Design Through Probabilistic Maximization". In: *Uncertainty in Complex Networked Systems: In Honor of Roberto Tempo*. Ed. by Tamer Baar. Cham: Springer International Publishing, 2018, pp. 247–274. ISBN: 978-3-030-04630-9. DOI: 10.1007/978-3-030-04630-9_7. URL: https://doi.org/10.1007/978-3-030-04630-9_7.

[76] Battista Biggio and Fabio Roli. "Wild patterns: Ten years after the rise of adversarial machine learning". In: *Pattern Recognition* 84 (2018), pp. 317–331.

[77] Inés Sittón Candanedo et al. "Machine Learning Predictive Model for Industry 4.0". In: *Knowledge Management in Organizations*. Ed. by Lorna Uden, Branislav Hadzima, and I-Hsien Ting. Cham: Springer International Publishing, 2018, pp. 501–510. ISBN: 978-3-319-95204-8.

[78] Nicholas Carlini and David Wagner. "Audio adversarial examples: Targeted attacks on speech-to-text". In: *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE. 2018, pp. 1–7.

[79] Arin Chaudhuri et al. "Sampling Method for Fast Training of Support Vector Data Description". In: *2018 Annual Reliability and Maintainability Symposium (RAMS)*. IEEE, Jan. 2018. DOI: 10.1109/ram.2018.8463127. URL: https://doi.org/10.1109%2Fram.2018.8463127.

[80] Sen Chen et al. "Automated poisoning attacks and defenses in malware detection systems: An adversarial machine learning approach". In: *computers & security* 73 (2018), pp. 326–344.

[81] K. Czarnecki and R. Salay. "Towards a Framework to Manage Perceptual Uncertainty for Safe Automated Driving". In: *International Workshop on Artificial Intelligence Safety Engineering (WAISE)* (2018).

[82] Amit Dhurandhar et al. "Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives". In: *NeurIPS*. 2018.

[83] Filip Doilovi, Mario Brcic, and Nikica Hlupic. "Explainable Artificial Intelligence: A Survey". In: May 2018. DOI: 10.23919/MIPRO.2018.8400040.

[84] Vasisht Duddu. "A survey of adversarial machine learning in cyber warfare". In: *Defence Science Journal* 68.4 (2018), p. 356.

[85] José Miguel Faria. "Machine Learning Safety: An Overview". In: *Safety-Critical Systems Club* (2018).

[86] Alessandro Fermi et al. "Identification of safety regions in vehicle platooning via machine learning". In: June 2018, pp. 1–4. DOI: 10.1109/WFCS.2018.8402372.

[87] Alessandro Fermi et al. "Identification of safety regions in vehicle platooning via machine learning". In: *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*. 2018, pp. 1–4. DOI: 10.1109/WFCS.2018.8402372.

[88] Riccardo Guidotti et al. "A Survey of Methods for Explaining Black Box Models". In: *ACM Comput. Surv.* 51.5 (Aug. 2018). ISSN: 0360-0300. DOI: 10.1145/3236009. URL: https://doi.org/10.1145/3236009.

[89] Bojan Kolosnjaji et al. "Adversarial malware binaries: Evading deep learning for malware detection in executables". In: *2018 26th European signal processing conference (EUSIPCO)*. IEEE. 2018, pp. 533–537.

[90] Siddique Latif, Rajib Rana, and Junaid Qadir. "Adversarial machine learning and speech emotion recognition: Utilizing generative adversarial networks for robustness". In: *arXiv preprint arXiv:1811.11402* (2018).

[91] Peter Mills. *Solving for multi-class: a survey and synthesis*. 2018. DOI: 10.48550/ARXIV.1809.05929. URL: https://arxiv.org/abs/1809.05929.

[92] Md Ashraful Alam Milton. "Evaluation of momentum diverse input iterative fast gradient sign method (M-DI2-FGSM) based attack method on MCS 2018 adversarial attacks on black box face recognition system". In: *arXiv preprint arXiv:1806.08970* (2018).

[93] M. Mongelli et al. "Performance validation of vehicle platooning via intelligible analytics". In: *IET Cyber-Physical Systems: Theory & Applications. 4. 10.1049/iet-cps.2018.5055* (2018).

[94] Maria-Irina Nicolae et al. "Adversarial Robustness Toolbox v1.2.0". In: *CoRR* 1807.01069 (2018). URL: https://arxiv.org/pdf/1807.01069.

[95] Andy Shih, Arthur Choi, and Adnan Darwiche. *A Symbolic Approach to Explaining Bayesian Network Classifiers*. July 2018. DOI: 10.24963/ijcai.2018/708. URL: http://dx.doi.org/10.24963/ijcai.2018/708.

[96] Takuya Akiba et al. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 2623–2631.

[97] Teodoro Alamo et al. "Safe approximations of chance constrained sets by probabilistic scaling". In: *2019 18th European Control Conference (ECC)*. 2019, pp. 1380–1385. DOI: 10.23919/ECC.2019.8795669.

[98] Luca Demetrio et al. "Explaining vulnerabilities of deep learning to adversarial malware binaries". In: *arXiv preprint arXiv:1901.03583* (2019).

[99] Samuel G Finlayson et al. "Adversarial attacks on medical machine learning". In: *Science* 363.6433 (2019), pp. 1287–1289.

[100] Sachin Grover et al. "BEEF: Balanced English Explanations of Forecasts". In: *IEEE Transactions on Computational Social Systems* 6.2 (2019), pp. 350–364. DOI: 10.1109/TCSS.2019.2902490.

[101] Riccardo Guidotti et al. "Factual and Counterfactual Explanations for Black Box Decision Making". In: *IEEE Intelligent Systems* 34.6 (2019), pp. 14–23. DOI: 10.1109/MIS.2019.2957223.

[102] Olakunle Ibitoye, Omair Shafiq, and Ashraf Matrawy. "Analyzing adversarial attacks against deep learning for intrusion detection in IoT networks". In: *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2019, pp. 1–6.

[103] Yuxin Ma et al. "Explaining vulnerabilities to adversarial machine learning through visual analytics". In: *IEEE transactions on visualization and computer graphics* 26.1 (2019), pp. 1075–1085.

[104] Nuno Martins et al. "Analyzing the footprint of classifiers in adversarial denial of service contexts". In: *EPIA Conference on Artificial Intelligence*. Springer. 2019, pp. 256–267.

[105] Maurizio Mongelli et al. "Accellerating PRISM Validation of Vehicle Platooning Through Machine Learning". In: Nov. 2019, pp. 452–456. DOI: 10.1109/ICSRS48664.2019.8987672.

[106] Maurizio Mongelli et al. "Accellerating PRISM Validation of Vehicle Platooning Through Machine Learning". In: *2019 4th International Conference on System Reliability and Safety (ICSRS)*. 2019, pp. 452–456. DOI: 10.1109/ICSRS48664.2019.8987672.

[107] Maurizio Mongelli et al. "Performance validation of vehicle platooning through intelligible analytics". In: *IET Cyber-Physical Systems: Theory & Applications* 4.2 (2019), pp. 120–127.

[108] Jiangmiao Pang et al. "Libra R-CNN: Towards Balanced Learning for Object Detection". In: *CoRR* abs/1904.02701 (2019). arXiv: 1904.02701. URL: http://arxiv.org/abs/1904.02701.

[109] Shilin Qiu et al. "Review of Artificial Intelligence Adversarial Attack and Defense Technologies". In: *Applied Sciences* 9.5 (2019). ISSN: 2076-3417. DOI: 10.3390/app9050909. URL: https://www.mdpi.com/2076-3417/9/5/909.

[110] Cynthia Rudin. *Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead*. 2019. arXiv: 1811.10154 [stat.ML].

[111] Yalin E Sagduyu, Yi Shi, and Tugba Erpek. "IoT network security from the perspective of adversarial deep learning". In: *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2019, pp. 1–9.

[112] Pinjie Sun and Liye Zhang. "Public Opinion Guidance Under the Background of Big Data Technology". In: *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*. 2019, pp. 226–226. DOI: 10.1109/ISI.2019.8823466.

[113] Emanuele Albini et al. *Relation-Based Counterfactual Explanations for Bayesian Network Classifiers*. July 2020. DOI: 10.24963/ijcai.2020/63. URL: http://dx.doi.org/10.24963/ijcai.2020/63.

[114] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2019.12.012. URL: https://www.sciencedirect.com/science/article/pii/S1566253519308103.

[115] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2019.12.012. URL: https://www.sciencedirect.com/science/article/pii/S1566253519308103.

[116] A. Campagner, F. Cabitza, and D. Ciucci. "Three-way decision for handling uncertainty in machine learning: a narrative review". In: *International Joint Conference on Rough Sets* (2020).

[117] Luca Demetrio et al. "Waf-a-mole: evading web application firewalls through adversarial machine learning". In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. 2020, pp. 1745–1752.

[118] Andrew Hryniowski, Xiao Yu Wang, and Alexander Wong. "Where Does Trust Break Down? A Quantitative Trust Analysis of Deep Neural Networks via Trust Matrix and Conditional Trust Densities". In: *CoRR* abs/2009.14701 (2020). arXiv: 2009.14701. URL: https://arxiv.org/abs/2009.14701.

[119] Zhengping Luo et al. "Adversarial machine learning based partial-model attack in IoT". In: *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*. 2020, pp. 13–18.

[120] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. "Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations". In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. FAT* '20. Barcelona, Spain: Association for Computing Machinery, 2020, pp. 607–617. ISBN: 9781450369367. DOI: 10.1145/3351095.3372850. URL: https://doi.org/10.1145/3351095.3372850.

[121] Daniel Nemirovsky et al. "CounteRGAN: Generating Realistic Counterfactuals with Residual Generative Adversarial Nets". In: *ArXiv* abs/2009.05199 (2020).

[122] AKM Newaz et al. "Adversarial Attacks to Machine Learning-Based Smart Healthcare Systems". In: *arXiv preprint arXiv:2010.03671* (2020).

[123] Rafael Poyiadzi et al. "FACE: Feasible and Actionable Counterfactual Explanations". In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (2020).

[124] Adnan Qayyum et al. "Secure and robust machine learning for healthcare: A survey". In: *arXiv preprint arXiv:2001.08103* (2020).

[125] Holger Trittenbach, Klemens Böhm, and Ira Assent. *Active Learning of SVDD Hyperparameter Values*. 2020. DOI: 10.1109/DSAA49011.2020.00023.

[126] Mehmet Turkoz et al. "Generalized support vector data description for anomaly detection". In: *Pattern Recognition* 100 (2020), p. 107119. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2019.107119. URL: https://www.sciencedirect.com/science/article/pii/S0031320319304200.

[127] Ivan Vaccari et al. "MQTTset, a New Dataset for Machine Learning Techniques on MQTT". In: *Sensors* 20.22 (2020), p. 6578.

[128] Giulia Vilone and Luca Longo. "Explainable Artificial Intelligence: a Systematic Review". In: *CoRR* abs/2006.00093 (2020). arXiv: 2006.00093. URL: https://arxiv.org/abs/2006.00093.

[129] Adam White and Artur S. d'Avila Garcez. "Measurable Counterfactual Local Explanations for Any Classifier". In: *ECAI*. 2020.

[130] Han Xu et al. "Adversarial attacks and defenses in images, graphs and text: A review". In: *International Journal of Automation and Computing* 17.2 (2020), pp. 151–178.

[131] Yangyang Zhao et al. "Novel Three-Stage Framework for Prioritizing and Selecting Feature Variables for Short-Term Metro Passenger Flow Prediction". In: *Transportation Research Record* 2674.8 (2020), pp. 192–205. DOI: 10.1177/0361198120926504. eprint: https://doi.org/10.1177/0361198120926504. URL: https://doi.org/10.1177/0361198120926504.

[132] Anastasios N. Angelopoulos and Stephen Bates. "A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification". In: *CoRR* abs/2107.07511 (2021). arXiv: 2107.07511. URL: https://arxiv.org/abs/2107.07511.

[133] Eirini Anthi et al. "Hardening machine learning denial of service (DoS) defences against adversarial attacks in IoT smart home networks". In: *computers & security* (2021), p. 102352.

[134] Vaishak Belle and Ioannis Papantonis. "Principles and Practice of Explainable Machine Learning". In: *Frontiers in Big Data* 4 (2021). ISSN: 2624-909X. DOI: 10.3389/fdata.2021.688969. URL: https://www.frontiersin.org/articles/10.3389/fdata.2021.688969.

[135] Alberto Carlevaro and Maurizio Mongelli. "A New SVDD Approach to Reliable and eXplainable AI". In: *IEEE Intelligent Systems* in press (2021), pp. 1–1. DOI: 10.1109/MIS.2021.3123669.

[136] Alberto Carlevaro and Maurizio Mongelli. "Reliable AI Through SVDD and Rule Extraction". In: Aug. 2021, pp. 153–171. ISBN: 978-3-030-84059-4. DOI: 10.1007/978-3-030-84060-0_10.

[137] Luca Demetrio et al. "Functionality-preserving black-box optimization of adversarial windows malware". In: *IEEE Transactions on Information Forensics and Security* (2021).

[138] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. "A Comprehensive Survey and Performance Analysis of Activation Functions in Deep Learning". In: *CoRR* abs/2109.14545 (2021). arXiv: 2109.14545. URL: https://arxiv.org/abs/2109.14545.

[139] Benyamin Ghojogh et al. *Reproducing Kernel Hilbert Space, Mercer's Theorem, Eigenfunctions, Nyström Method, and Use of Kernels in Machine Learning: Tutorial and Survey*. 2021. arXiv: 2106.08443 [stat.ML].

[140] Wei Guo et al. "Multi-kernel Support Vector Data Description with boundary information". In: *Engineering Applications of Artificial Intelligence* 102 (2021), p. 104254. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2021.104254. URL: https://www.sciencedirect.com/science/article/pii/S0952197621001019.

[141] Hui Hou and Hongquan Ji. "Improved multiclass support vector data description for planetary gearbox fault diagnosis". In: *Control Engineering Practice* 114 (2021), p. 104867. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2021.104867. URL: https://www.sciencedirect.com/science/article/pii/S0967066121001441.

[142] Mongelli Maurizio and Orani Vanessa. "Stability Certification of Dynamical Systems: Lyapunov Logic Learning Machine". In: July 2021, pp. 221–235. ISBN: 978-981-33-6172-0. DOI: 10.1007/978-981-33-6173-7_15.

[143] Marco Mirabilio et al. "String Stability of a Vehicular Platoon With the Use of Macroscopic Information". In: *IEEE Transactions on Intelligent Transportation Systems* 22.9 (2021), pp. 5861–5873. DOI: 10.1109/TITS.2021.3056237.

[144] Rami Mochaourab et al. "Robust Counterfactual Explanations for Privacy-Preserving SVM." In: 2021.

[145] M. Mongelli. "Design of countermeasure to packet falsification in vehicle platooning by explainable artificial intelligence". In: *Computer Communications* 179 (2021), pp. 166–174. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2021.06.026. URL: https://www.sciencedirect.com/science/article/pii/S0140366421002504.

[146] M. Mongelli. "Design of countermeasure to packet falsification in vehicle platooning by explainable artificial intelligence". In: *Computer Communications* (2021). ISSN: 0140-3664. URL: https://www.sciencedirect.com/science/article/pii/S0140366421002504.

[147] S. Narteni et al. "From Explainable to Reliable Artificial Intelligence". In: *International IFIP Cross Domain (CD) Conference for Machine Learning & Knowledge Extraction (MAKE), CD-MAKE 2021.* (2021).

[148] Daniel Nemirovsky et al. "Providing Actionable Feedback in Hiring Marketplaces Using Generative Adversarial Networks". In: WSDM '21. Virtual Event, Israel: Association for Computing Machinery, 2021. ISBN: 9781450382977. DOI: 10.1145/3437963.3441705. URL: https://doi.org/10.1145/3437963.3441705.

[149] Yulexis Pacheco and Weiqing Sun. "Adversarial Machine Learning: A Comparative Study on Contemporary Intrusion Detection Datasets." In: *ICISSP*. 2021, pp. 160–171.

[150] Iqbal H Sarker. "Machine learning: Algorithms, real-world applications and research directions". In: *SN Computer Science* 2.3 (2021), pp. 1–21.

[151] Muneaki Suzuki et al. "Understanding the Reason for Misclassification by Generating Counterfactual Images". In: *2021 17th International Conference on Machine Vision and Applications (MVA)*. 2021, pp. 1–5. DOI: 10.23919/MVA51890.2021.9511352.

[152] Ivan Vaccari et al. "A Generative Adversarial Network (GAN) Technique for Internet of Medical Things Data". In: *Sensors* 21.11 (2021). ISSN: 1424-8220. URL: https://www.mdpi.com/1424-8220/21/11/3726.

[153] Arnaud Van Looveren and Janis Klaise. *Interpretable Counterfactual Explanations Guided by Prototypes*. 2021. DOI: 10.1007/978-3-030-86520-7_40. URL: http://dx.doi.org/10.1007/978-3-030-86520-7_40.

[154] Michael Veale and Frederik Zuiderveen Borgesius. "Demystifying the Draft EU Artificial Intelligence Act Analysing the good, the bad, and the unclear elements of the proposed approach". In: *Computer Law Review International* 22.4 (Aug. 2021), pp. 97–112. ISSN: 2194-4164. DOI: 10.9785/cri-2021-220402. URL: http://dx.doi.org/10.9785/cri-2021-220402.

[155] Giulia Vilone and Luca Longo. "Classification of Explainable Artificial Intelligence Methods through Their Output Formats". In: *Machine Learning and Knowledge Extraction* 3.3 (2021), pp. 615–661. ISSN: 2504-4990. DOI: 10.3390/make3030032. URL: https://www.mdpi.com/2504-4990/3/3/32.

[156] Giulia Vilone and Luca Longo. "Classification of Explainable Artificial Intelligence Methods through Their Output Formats". In: *Mach. Learn. Knowl. Extr.* 3 (2021), pp. 615–661. URL: https://api.semanticscholar.org/CorpusID:238811354.

[157] Alexander Wong, Xiao Yu Wang, and Andrew Hryniowski. *How Much Can We Really Trust You? Towards Simple, Interpretable Trust Quantification Metrics for Deep Neural Networks*. 2021. arXiv: 2009.05835 [cs.LG].

[158] Abdurrouf et al. "The Seventeenth Data Release of the Sloan Digital Sky Surveys: Complete Release of MaNGA, MaStar, and APOGEE-2 Data". In: *The Astrophysical Journal Supplement Series* 259.2 (2022), p. 35. DOI: 10.3847/1538-4365/ac4414. URL: https://dx.doi.org/10.3847/1538-4365/ac4414.

[159] Alberto Carlevaro et al. "Counterfactual Building and Evaluation via eXplainable Support Vector Data Description". In: *IEEE Access* 10 (2022), pp. 60849–60861. DOI: 10.1109/ACCESS.2022.3180026.

[160] Enrico Ferrari et al. "A Novel Rule-Based Modeling and Control Approach for the Optimization of Complex Water Distribution Networks". In: *Advances in System-Integrated Intelligence: Proceedings of the 6th International Conference on System-Integrated Intelligence (SysInt 2022), September 7-9, 2022, Genova, Italy*. Springer. 2022, pp. 33–42.

[161] Riccardo Guidotti. "Counterfactual explanations and how to find them: literature review and benchmarking". en. In: *Data Mining and Knowledge Discovery* (2022). DOI: 10.1007/s10618-022-00831-6. URL: http://dx.doi.org/10.1007/s10618-022-00831-6.

[162] Glenn Jocher et al. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation*. Version v7.0. Nov. 2022. DOI: 10.5281/zenodo.7347926. URL: https://doi.org/10.5281/zenodo.7347926.

[163] Marta Lenatti et al. "A novel method to derive personalized minimum viable recommendations for type 2 diabetes prevention based on counterfactual explanations". en. In: *PLOS ONE* 17.11 (2022). Ed. by Muhammad Fazal Ijaz. DOI: 10.1371/journal.pone.0272825. URL: http://dx.doi.org/10.1371/journal.pone.0272825.

[164] Marta Lenatti et al. "Characterization of Type 2 Diabetes using Counterfactuals and Explainable AI". In: *Proceedings of the 32nd Medical Informatics Europe (EFMI MIE 2022) Conference, May 27-30, 2022, Nice, France. Published in Studies in Health Technology and Informatics* 294 (May 2022), pp. 98–103. DOI: 10.3233/shti220404. URL: http://dx.doi.org/10.3233/SHTI220404.

[165] Martina Mammarella et al. "Chance-constrained sets approximation: A probabilistic scaling approach". In: *Automatica* 137 (2022), p. 110108. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2021.110108. URL: https://www.sciencedirect.com/science/article/pii/S0005109821006373.

[166] Victor Mirasierra et al. "Prediction Error Quantification Through Probabilistic Scaling". In: *IEEE Control Systems Letters* 6 (2022), pp. 1118–1123. DOI: 10.1109/LCSYS.2021.3087361.

[167] Pablo Del Moral, Sawomir Nowaczyk, and Sepideh Pashami. "Why Is Multiclass Classification Hard?" In: *IEEE Access* 10 (2022), pp. 80448–80462. DOI: 10.1109/ACCESS.2022.3192514.

[168] Chao Zhai and Hung D. Nguyen. "Estimating the Region of Attraction for Power Systems Using Gaussian Process and Converse Lyapunov Function". In: *IEEE Transactions on Control Systems Technology* 30.3 (May 2022), pp. 1328–1335. ISSN: 1558-0865. DOI: 10.1109/TCST.2021.3098167.

[169] Alberto Carlevaro et al. "Multi-Class Counterfactual Explanations using Support Vector Data Description". In: (Mar. 2023). DOI: 10.36227/techrxiv.22221007.v1. URL: https://www.techrxiv.org/articles/preprint/Multi-Class_Counterfactual_Explanations_using_Support_Vector_Data_Description/22221007.

[170] Alberto Carlevaro et al. *Probabilistic Safety Regions Via Finite Families of Scalable Classifiers*. 2023. arXiv: 2309.04627 [stat.ML].

[171] A. Carlevaro and M. Mongelli. "A New SVDD Approach to Reliable and eXplainable AI". In: *IEEE Intelligent Systems* 01 (Oct. 5555), pp. 1–1. ISSN: 1941-1294. DOI: 10.1109/MIS.2021.3123669.

[172] F. Abramovich and Y. Ritov. *Statistical Theory: A Concise Introduction*. Chapman & Hall/CRC Texts in Statistical Science. CRC Press. ISBN: 9781482211849.

[173] *General Data Protection Regulation (GDPR)*. https://gdpr.eu/tag/gdpr/. [Retrieved December 16, 2022]. European Commission.

[174] Kools J. *6 functions for generating artificial datasets*. URL: https://www.mathworks.com/matlabcentral/fileexchange/41459-6-functions-for-generating-artificial-datasets. MATLAB Central File Exchange. Retrieved April 4, 2021.

[175] KEEL. *ebsite: KEEL (Knowledge Extraction based on Evolutionary Learning), W Nov. 2012. [Online]. Available:* URL: http://sci2s.ugr.es/keel/datasets.php.

[176] *Rulex Analytics platform, https://www.rulex.ai/*.

[177] Alberto Sassu et al. "Artichoke Deep Learning Detection Network for Site-Specific Agrochemicals Uas Spraying". In: *Available at SSRN 4272684* ().

[178] *Turbofan engine degradation simulation data set, https://data.nasa.gov/Aerospace/Turbofan-engine-degradation-simulation-data-set/vrks-gjie/*.

[179] *EASA Concept Paper: First usable guidance for Level 1 machine learning applications, a deliverable of the EASA AI Roadmap*. Standard. Konrad-Adenauer-Ufer 3 50668 Cologne Germany: European Union Aviation Safety Agency, Apr. 2021.

[180] *Concepts of Design Assurance for Neural Networks CoDANN*. Standard. Also available as yperrefhttps://www.easa.europa.eu/sites/default/files/dfu/EASA-DDLN-Concepts-of-Design-Assurance-for-Neural-Networks-CoDANN.pdf. Daedalean, AG: European Union Aviation Safety Angency, Mar. 2020.

Paddle your own canoe.