ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Istituto di Elettronica e di Ingegneria
dell'Informazione e delle Telecomunicazioni

Doctoral Dissertation
Doctoral Program in Artificial Intelligence - Industry 4.0 (37.th cycle)

Reliability Assessment Methods for eXplainable Artificial Intelligence

Sara Narteni
* * * * *

Supervisors
Dott. Maurizio Mongelli, CNR-IEIIT, Supervisor
Prof. Fabrizio Dabbene, CNR-IEIIT, Co-Supervisor

Referees
Prof. Serge Autexier, German Research Center for Artificial Intelligence
Prof. Maria Chiara Leva, Technological University Dublin

Politecnico di Torino
2025

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . .

Sara Narteni

Turin, 2025

Summary

The rapid evolution of machine learning (ML) algorithms in recent years, along with the availability of powerful technological infrastructures supporting them, is generating more and more interest in adopting artificial intelligence (AI) solutions in many domains. These also include safety-critical ones, such as healthcare, smart mobility or cybersecurity, which deserve a special attention. Trustworthy AI (TAI) framework has been introduced to individuate the requirements that AI systems should fulfill during all their design, development, and deployment stages, in order to be lawful, ethical, and robust. Among these, eXplainable AI (XAI) and Reliable AI (RAI) emerge as two key properties that must be guaranteed in pursuing a trustworthy AI solution. XAI collects a broad set of techniques providing insights to the logic behind AI-based decision-making systems. On the other hand, RAI consists in ensuring that such systems work with adequate performance both at the model level and at a system level. Current research on trustworthy AI recognizes the important role of XAI in reliability assessment, since a lack of the first compromises the latter. However, a completely unified vision on XAI and RAI is still little investigated.

In this context, this dissertation focuses on studying how to integrate reliability in rule-based binary classification methods, thus attempting to establish a link between XAI and RAI. The focus is posed on rule-based classifiers of the *if-then* kind, since they constitute, at least in principle, one of the most simple forms of XAI. Nevertheless, this is not always true, since rule-based classifiers often reveal to be sensitive to the complexity of the data under analysis, providing decision rules that are not so straightforward to comprehend, especially by non-experts of a given application domain. Therefore, the preliminary contribution of this dissertation is the introduction of innovative *rule similarity* metrics that allow to compare and extract knowledge from sets of rules. Specifically, three new metrics are introduced: syntactic rule similarity and Bag of Words similarity, which, through different mechanisms, both compare rules in terms of their syntax, and geometrical rule similarity, which accounts for how rules are geometrically related to each other in the feature space.

The research then moves to designing *rule-based safety regions*, i.e., regions of the feature space that ensure a guaranteed performance of the rule-based classifier on a

target class. Such target depends on the application scenario, and typically represents a safe situation, intended as the absence of conditions that can cause harms to humans/environments: for example, the absence of collisions in autonomous driving, or the absence of a pathology in healthcare applications. The final objective is that safety regions, representing the subset of inputs that will most probably lead to a correct performance of the rule-based classifier, can serve as monitoring tools over the inputs, possibly triggering alerts when these fall outside their boundaries. Yet pursuing the same objective, two different solutions to this problem are researched.

Initially, a heuristic approach is explored, where the regions are designed starting from the rules themselves - opportunely synthesized by their feature and value ranking properties - and optimizing their thresholds in a grid-search-like mode, until achieving the minimal statistical error on the desired class. These methods, called *reliability from inside*, *reliability from outside* and *rules with zero error*, look at identifying the safety regions by exploiting the properties of the rules for the target class, the non-target class, and a combination of target class rules specifically trained with zero error constraint, respectively. Despite the heuristic approach, experiments show promising results when tested in some safety-critical applications such as the collision avoidance in vehicle platooning scenarios, the prediction of physical fatigue, and also the detection of adversarial ML attacks.

Afterwards, a more formal solution is sought by relying on a well-established statistical framework, widely used in ML uncertainty quantification: *conformal prediction* (CP), which allows to probabilistically guarantee the error rate under a desired level, by assigning prediction sets to data samples, instead of the typical point predictions provided by classifiers. The key aspect of CP is the design of a score function that encodes the behavior of a classifier, assigning larger values to encode a worse agreement between a point and a candidate label, and vice versa. In this respect, a *novel score function*, called CONFIDERAI, that allows to apply CP theory to rule-based binary classifiers is proposed, accounting for both the geometrical structure (distance of points to rule boundaries, and rule overlaps), and the predictive performance of decision rules (i.e., rule relevance). Leveraging on the results provided through CONFIDERAI, the *conformal critical set* has been defined as the subset of points in the input space where the prediction set is solely composed by points belonging to the target class, and whose error is bounded by the CP. It is shown that this definition provides a new labelling of the dataset, which reveals useful for the generation of new rules that have improved performance on the target class with respects to the original ruleset. Extensive experimentation on both toy datasets/rulesets and real-world applications shows good results, highlighting the relevance of this contribution at the intersection of explainability and reliability.

Finally, these techniques are tested in some applications of interest in Industry 4.0, also deriving from research projects I contributed to.

Acknowledgements

Nel ripensare alla me stessa di ormai quasi 5 anni fa, appena laureata ed un po' "smarrita" rispetto alle diverse strade che potessi avere di fronte, mi piace sempre ricordare di come, quasi per caso, mi fossi affacciata al mondo della ricerca, intraprendendo questo percorso di dottorato. Compiere questa scelta ha significato per me non solo affrontare tematiche scientifiche quasi completamente nuove, imparando nuovi argomenti e soprattutto imparando ad apprenderli, ma mettermi alla prova con quanto di più avessi sempre ritenuto essere troppo al di là della mia zona di comfort: dal presentare le mie ricerche in pubblico, al partecipare ad eventi per "semplice" networking. Ed è così che questi 3 anni mi hanno permesso di crescere sotto il punto di vista professionale, trovando una direzione che la me neolaureata non sentiva di avere, ma anche dal punto di vista personale. Grazie al dottorato, posso dire - sono sincera, non senza una buona dose di orgoglio - di avere fatto un importante passo in avanti nel migliorare quelli che sono sempre stati un po' i miei limiti personali: avere parlato in pubblico ormai non so quante volte; avere partecipato a conferenze dove ho essenzialmente parlato con sconosciuti o di fronte ai guru degli argomenti che stavo studiando; di avere trascorso alcuni mesi all'estero, da sola, lontana da casa. E, perchè no, ci tengo anche a ricordare di tutte le belle opportunità che ho avuto di vedere nuovi posti nel mondo: dalla mia adorata Siviglia, a Lisbona, alla Bassa Sassonia, a Cipro e Malta e, non da ultimo, alla California. Chi mai avrebbe pensato di andare anche in America?!

Penso e spero che queste righe abbiano dato un quadro, seppur non esaustivo, di ciò che per me sono stati questi 3 anni. Ma niente di tutto quello che ho citato sarebbe stato possibile senza tutte le persone che mi hanno accompagnato in questo bel percorso (e che, almeno per ora, non si libereranno di me), spronandomi sempre in tutte le attività. Ci tengo quindi a ringraziare tutto il mio gruppo di ricerca.
Grazie ai miei supervisor, Maurizio Mongelli e Fabrizio Dabbene, per la loro importante guida scientifica, che mi hanno dato sempre le giuste motivazioni e permesso di avere tante belle opportunità.
Grazie ai miei compagni di avventura nonchè preziosi collaboratori: grazie ad Alberto, che mi ha aiutata a fare un pochino più mia la matematica e mi ha sempre supportato quando più ne avevo bisogno; grazie a Marta, con cui ho condiviso tante

belle esperienze, soprattutto nella nostra “terra d’origine” dell’ingegneria biomedica. Grazie anche ai collaboratori “passati”, Giacomo, Ivan e Vanessa, che hanno contribuito con me ad alcuni lavori che oggi presento in questa tesi.
E grazie infine a tutti gli altri colleghi di Genova, “quelli dei pranzi belli’: Alessandra, Barbara, Enrico (a cui va la menzione speciale per avermi avviato al python e no, non rinnego il passato cyber), Martina, Sabrina, Sandro, Paola, con cui le risate non mancano mai.

Non posso poi non ringraziare la mia famiglia: i miei genitori, che mi hanno aiutato tanto in questi anni, permettendomi di essere giunta a questo importante traguardo; mia sorella Elena, che - seppur a suo modo - sa sempre come rincuorarmi e che, finalmente, adesso non mi sentirà più dire “I’m Sara Narteni, a PhD student,...” mentre provo le presentazioni. Grazie ai miei nonni, Anna, Armando e Mariuccia, che per me ci sono sempre, anche se per loro il funzionamento di un dottorato rimarrà un mistero maggiore di quelli che si nascondono dietro all’Intelligenza Artificiale ;-).

Non so bene cosa mi riserverà il futuro, ma... *Ad Maiora!*

Sara

"La vita non è facile per nessuno di noi.
Ma cosa importa? Bisogna perseverare e,
soprattutto, avere fiducia in sé stessi.
Bisogna credere di essere dotati per
qualcosa e che questa cosa deve essere
raggiunta."
M. Curie

Alla mia famiglia.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This Chapter introduces the context that this dissertation addresses. It gradually moves from introducing the main principles of ***trustworthy AI*** (Section 1.1), which is the broader scope of this thesis, then it gives an overview on the origins and meanings of ***reliability*** assessment in the context of AI (Section 1.2), showing how this can include both an "internal" model-level perspective, faced by uncertainty quantification, and an "external" system-level perspective, studied through standards and regulations. An introduction to the need of more research towards integrating ***explainability*** into reliable AI is provided in Section 1.3, and Section 1.4 describes how this dissertation contributes to this, highlighting the research questions and the main approaches studied in my PhD. Finally, Section 1.5 describes how the remaining of the thesis is structured.

1.1 Trustworthy Artificial Intelligence paradigm

The quick advancement of ML algorithms capabilities and the availability of computational resources have boosted interest in the opportunities offered by such technologies in many domains. ML/AI-based solutions are indeed spreading as promising support tools for tackling real-world problems in several fields of industry and, more generally, of society [57]. These also include safety-critical sectors like healthcare, transports or avionics, where ensuring that any ML-based decision-making is performed as intended is essential to prevent harmful consequences for the people and/or the environment where the AI system acts.
It is in this context that the *Trustworthy Artificial Intelligence* framework has been introduced by the European Commission High-Level Expert Group on AI in 2019 [60]. TAI collects all the requirements that an AI system must fulfill to be developed, deployed and used in real-world, potentially sensitive, applications and envisions AI systems that are human-centred, trying to maximise the benefits while preventing and reducing the risks. TAI is founded on three pillars: 1) *lawfulness*,

i.e., complying with regulation; 2) *ethics*, i.e., ensuring adherence to ethical principles; 3) *robustness*, i.e., AI systems should perform in a safe, secure and reliable manner, and safeguards should be foreseen to prevent unintended effects.

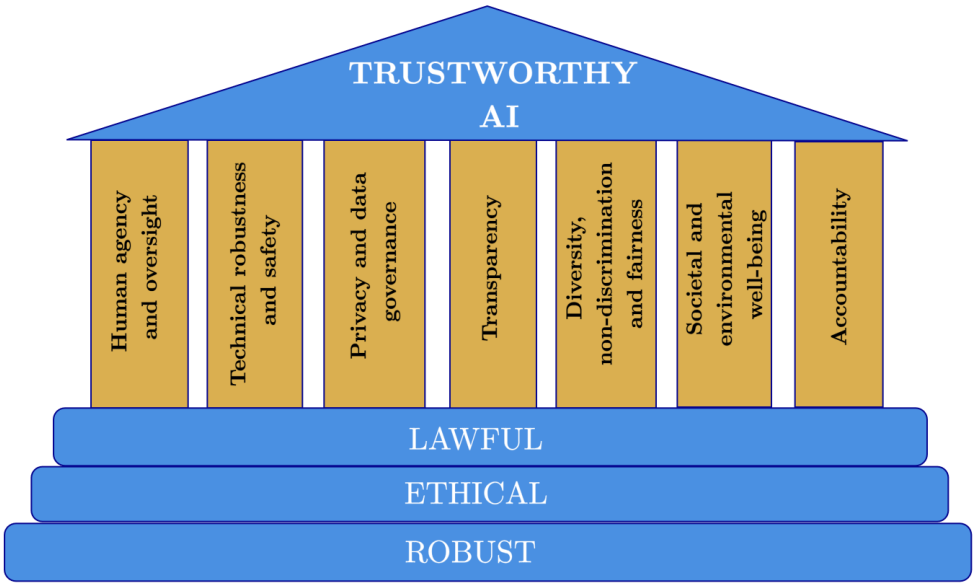


Figure 1.1: Principles of Trustworthy AI (adapted from [36]).

On the basis of these three concepts, seven requirements are identified, as shown in Figure 1.1:

1. *Human Agency and Oversight*: AI should empower human autonomy and decision-making, ensuring that individuals remain in control and can intervene when needed.

2. *Technical Robustness and Safety*: AI systems must be *reliable*, secure, and resilient. They need to operate safely under various conditions and handle unexpected inputs or failures, minimizing risks of harm.

3. *Privacy and Data Governance*: AI systems must be privacy-preserving and must handle sensitive data properly.

4. *Transparency*: AI operations should be transparent and understandable to users and stakeholders, who need to be informed about how decisions are made. This involves clear documentation, *explainability* of models, and access to system information.

5. *Diversity, Non-Discrimination, and Fairness*: AI should be inclusive and equitable, ensuring that biases are identified and mitigated.

6. *Societal and Environmental Well-being*: AI Systems should promote common good and contribute positively to environmental sustainability, public health, and overall societal welfare.

7. *Accountability*: Developers, operators, and users of AI systems must be accountable for the outcomes. Clear mechanisms for addressing harmful or unethical AI practices are necessary to maintain trust and responsibility in AI development and application.

The implementation of these requirements should occur throughout an AI system's entire life cycle and depends on the specific application.
The recent introduction of the European Union AI ACT [42] brings TAI to enclose an even broader "risk-based" vision, where the obligations related to the use of AI depend on the level of risk. Four risk levels for AI systems are individuated [36]: 1) Minimal or no risk applications, which can be used freely by following voluntary codes of conduct; 2) Limited risk applications, which need to adhere to specific transparency requirements, so that the users are always informed about their interaction with the AI; 3) High-risk applications, whose failure can create adverse impacts on people's safety or their fundamental rights, and thus must follow stringent regulation and assessments; 4) Unacceptable risk, that refers to those AI systems representing a clear threat to the lives and rights of people and that the AI Act prohibits their in the EU market. .

In this landscape, research on AI is currently expanding its horizons to address the requirements established by TAI. The following sections will provide definitions and background on the two AI properties tackled by this thesis, reliability and explainability, that are important parts of TAI *technical robustness and safety* and *transparency* principles.

1.2 Reliability in the AI context

Common performance metrics traditionally adopted to evaluate ML models on test data, such as accuracy in classification tasks, do not always reflect the way models perform in practice, especially when fed with slightly different or more challenging real-world data [112]. And this cannot be accepted to provide trustworthy solutions, especially in safety-critical scenarios.
In this context, *reliable AI* (RAI) emerges as a subfield of AI techniques addressing the challenge of providing the necessary performance guarantees to ML models, thus being an essential element in trustworthy AI. The term *reliability*, now widely used in the AI sphere, has a long tradition in the field of reliability engineering, which is the discipline born to ensure that a product or system performs as intended, without failing or - at least - staying within some specified performance limits [122].

Here, it is defined as "*the ability of a system or component to perform its required functions under stated conditions for a specified period of time*" [65].
In statistics and psychometrics, reliability of a measurement is linked to its repeatability and overall consistency, meaning that a measure is deemed reliable when it gives similar results under consistent conditions [89]. From measurement theory, it is known that a measurement is composed by two factors: a true score and a random error score that brings uncertainty; in this perspective, reliability provides an indication of the relative influence of true and error scores on the measurement, and can only be *estimated* since there is no direct way to access the true score [114]. Based on this, a clear link between reliability and uncertainty quantification (UQ) emerges, with the rise of many UQ techniques, as I briefly outline in Sec. 1.2.2.

The concept of reliability in the ML/AI world has been developed by drawing on these multidisciplinary definitions. According to the European guidelines on trustworthy AI [60], reliable AI contributes to the technical robustness and safety principle, by ensuring that an AI system *"works properly with a range of inputs and in a range of situations"*. Reliability is linked to safety in that it guarantees that the AI performs as expected, which is critical for preventing accidents, reducing errors, and avoiding harmful outcomes. In this respect, certification of the AI design and deployment constitutes another way to look at reliability.
Furthermore, the United States Artificial Intelligence Risk Management Framework [133] also recognized reliability as a pillar for enhancing trustworthiness of AI systems, and defines it as *"the overall correctness of AI system operation under the conditions of expected use and over a given period of time, including the entire lifetime of the system"*. Reliable AI is therefore devoted to ensure the continuity of accurate service by a ML model over time, also providing high confidence in its outcomes [121].
In 2009, authors of [20] provided a specific definition of reliability in ML as the "*qualitative property or ability of the system which is related to a critical performance indicator (positive or negative) of that system [...]*" and highlighted the importance of reliability estimation at the individual prediction level when dealing with risk-sensitive decision-making.
Ten years later, other researchers [122] identified three principles of reliability engineering useful to guide technical solutions for guaranteeing reliability in machine learning systems: i) failure prevention, i.e., developing ML algorithms that avoid possible incorrect predictions; ii) monitoring and failure identification at runtime; iii) maintenance of the model, i.e. fixing or preventing failures when they occur.
In the spirit of the definitions from reliability engineering, but with the goal of moving to a notion of *general reliability*, authors in [138] consider an AI model reliable if it meets three desiderata: i) uncertainty handling capabilities, ii) robust generalization to new scenarios, and iii) efficient adaptation to new data. Moreover, [33] identifies reliability as the conjunction of in-distribution performance (i.e.,

on unseen data coming from the same distribution as the training data), robustness, calibration, and out-of-distribution detection, and proposes a unified metric to assess state-of-the-art ML techniques.
In summary, it appears clear how reliable AI is really a multifaceted topic, and no unique definition is provided to date. In practice, reliability can be mainly studied from two perspectives: from one side, the "external" reliability assessment, strictly related to safety engineering, carried out by legislation and standardization efforts that act at a system level (Sec. 1.2.1 provides a short overview on this); from the other side, the "internal" reliability assurance, covering all methods aimed to quantify uncertainty and provide performance guarantees to ML models through suitable interventions made at some stage of the ML pipeline itself (see Sec. 1.2.2).

1.2.1 Certification of AI

From a safety engineering viewpoint, reliability can be seen as a prerequisite for AI safety assurance: a system that is not reliable, i.e. that produces frequent errors, behaves unpredictably, or is not sufficiently robust under given conditions, cannot be considered safe, since it cannot avoid unintended harmful consequences due to AI errors [34]. In light of this, research on reliable AI should not disregard the progresses and efforts that nowadays are being put towards the *certification of AI* algorithms, with many guidances and standards arising.
Traditionally, the certification of safety-critical systems has been covered by standards like IEC 61508 for electric/electronic systems [64] and related domain-specific adaptations (e.g., ISO26262 for automotive [68], DO-178C for avionics [158], or ISO 14971 for medical devices risk management [67]). Also, the topic has been studied from a total safety management viewpoint, which connects safety assessment with other business processes that may be used for quality, productivity and design [75]. The objective is to provide guidelines, risk-level and a range of requirements that need to be enforced in order to reach a certain level of safety based on the criticality of the system and/or components inside the whole architecture [134]. However, the introduction of AI components in these systems, and the subsequent increase of complexity levels, brings the need of rethinking the existing standards in light of the characteristics of the AI algorithms.
Starting from 2018, the international Joint Technical Committee 1 in Subcommittee 42 (JTC 1/SC 42) of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) has been developing standards for the AI ecosystem (31 published and 36 under development as of August 2024), including some dedicated to trustworthiness (e.g., ISO/IEC TR 24028:2020) and a guidance to AI quality evaluation (ISO/IEC TS 25058:2024) that also tackles reliability [69]. Besides the work from institutions/standardization committees, research groups are also proposing their solutions, such as the Assurance of Machine Learning for use in Autonomous Systems (AMLAS) developed by the University

Of York (UK), which is based on six stages devoted to systematically integrating safety assurance into the development of ML components [58].
Moreover, field-specific AI regulation is arising. A good example comes from the avionics sector, where the traditional approach to software certification that relied on the Verification and Validation (V-model) [91] has been redesigned by the European Union Aviation Safety Agency (EASA) in the Concepts of Design Assurance for Neural Networks (CoDANN) [39] to handle the data-driven nature of ML systems, giving rise to the so-called W-model (Fig. 1.2).

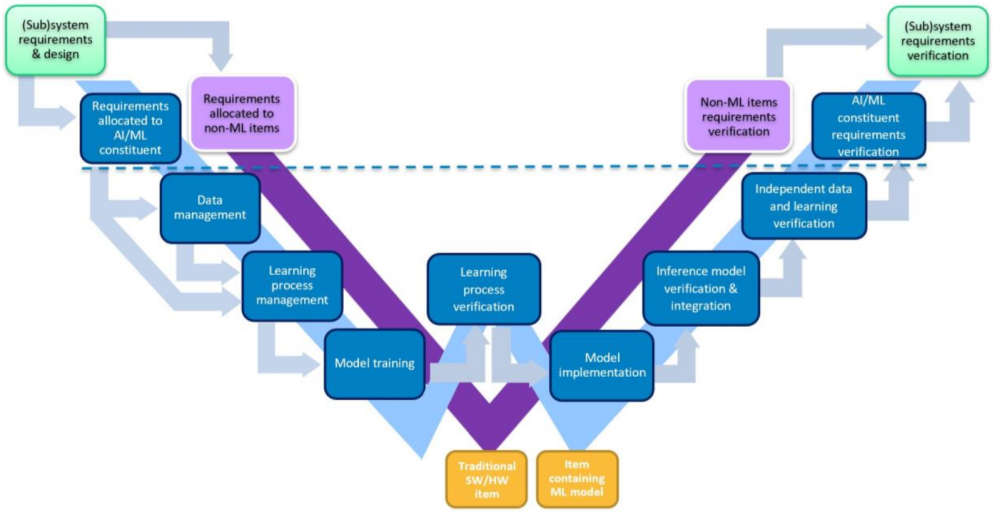


Figure 1.2: W-shaped Learning Assurance process superimposed on classical non-AI component V cycle process [39]. The dotted blue line separates the phases covered by traditional approaches (requirements management and verification, above the line) and those where an adaptation to the data-driven learning approaches is required (below the line).

Through this model, it is possible to achieve what EASA defined *learning assurance*, i.e., *"all of those planned and systematic actions used to substantiate, at an adequate level of confidence, that errors in a data-driven learning process have been identified and corrected such that the system satisfies the applicable requirements at a specified level of performance, and provides sufficient generalisation and robustness guarantees"*. Involving the concepts of generalisation and robustness, this definition points out once again the importance of reliability in AI. W-model envisions to go through several key steps to accomplish learning assurance at the different stages of the ML pipeline: data management, learning algorithm design, model training, learning verification, implementation and subsequent operational-time verification. RAI methods, including those being subject of this dissertation (see 1.4 and Chapter 4), can be considered part of the learning process verification,

located at the center of the W-model (Fig. 1.2). As CoDANN states, any shortcoming raised in this phase often leads the need to iterate again on the previous steps. From this point of view, RAI provides quantitative solutions (e.g., through the achievement of a given statistical error or confidence level) that constitute the output of the first iteration of the verification process, resulting in a useful tool to support the subsequent verification process phases. In practice, this may reduce the number of iterations required throughout the W-model's left branch, improving and simplifying the overall safety engineering process. For example, data-driven safety regions (both rule-based, Chapter 4, and black-box ones, see 5.3 and A.2.2) individuate where, in the feature space, the optimal ML performance is achieved. This may imply the need to act at the data management level, that is, by monitoring if the data lie in those regions or not. If not, fallback plans such as new training configurations might be necessary before passing the verification phase successfully. On the other hand, when data is compliant with the safety regions, the success of the learning verification should be guaranteed and the process can go on with the next phases.

Another prominent example of certification effort is the ISO/PAS 21448:2022 Safety of the intended functionality (SOTIF) from the automotive field [70], which can be considered the most mature in AI assurance. It introduces the concept of *intended* functionality, thus considering the case that a hazard may occur even in absence of a system failure, but rather when the system faces an unexpected scenario. Each scenario is categorized by SOTIF into one of the following four areas:

- *Known Safe (Area 1)*: scenarios where the system operates following its nominal behavior;
- *Known Unsafe (Area 2)*: scenarios whose limitations can be identified and understood well;
- *Unknown Unsafe (Area 3)*: also referred to as "*black swans*", these scenarios are the primary concern within SOTIF, since they constitute those cases that the system has not encountered before, and was not tested on, therefore leading to potentially unsafe behaviors;
- *Unknown Safe (Area 4)*: scenarios that the system has not encountered before but could handle safely if they arise.

Known safe scenarios (area 1) typically fall within the Operational Design Domain (ODD) of the system, that is the collection of conditions under which the system is designed to function[120]. A thorough characterization of the ODD and methods for acknowledging when scenarios leave it are essential to AI safety[13]. The goal of SOTIF realizes through the maximization of area 1 and the minimization of area 2 and 3 during development.Minimizing area 2 involves handling edge situations that

the system is aware of, although not being fully able to handle yet, thus requiring to consider fallback plans (e.g., taking over of human control). Minimizing area 3, in contrast, involves the more challenging task of discovering unforeseeable novel situations that might result into hazardous behaviors of the automated system, in a context denoted by high unpredictability. Overall, two stages can be identified within SOTIF process [62]: i) moving from unknown unsafe scenarios into known unsafe scenarios (i.e., from area 3 to area 2), which requires the identification of unknown scenarios; ii) transforming known unsafe scenarios into known safe scenarios by analyzing the identified hazardous scenarios and implementing measures to mitigate their associated risks. These stages are closely related to the ODD as well, since the ODD evolves with the system: it can be reduced (for limited time) to pursue safety guarantees (e.g., speed reductions in bad weather conditions), or it can be expanded when the system's robustness increases [48].
It appears then clear how all these points raised by SOTIF, and standardization in general, well relate to reliable AI, which, from this engineering point of view, can be seen as the set of techniques that drive the challenging tasks of ODD characterization, and the search and subsequent reduction of edge cases and black swans [74]. Uncertainty quantification plays an important role in solving these problems by working on the ML pipeline itself.
These considerations also outline the actual link existing between system-level and model-level reliability, an example of which can be found in the safety analysis of REXASI-PRO European project I contributed to, reported in Appendix A.1.

1.2.2 Uncertainty quantification

Sophisticated ML/DL models (e.g., neural networks), despite their very high predictive accuracy, often suffer from a lack of generalizability to unseen situations that compromises their proper functioning when deployed. Therefore, quantifying their predictive uncertainty is vital and is an important pillar in reliable AI. In this context, uncertainty can be defined as *"any deviation from the unachievable ideal of completely deterministic knowledge of the relevant system"* [144]. ML models indeed operate with incomplete knowledge (due to limited data, model limitations, etc.), leading to uncertainty in their predictions. The goal of *uncertainty quantification* approaches in ML is thus to reduce this deviation as much as possible, also acknowledging that some level of uncertainty is inevitable. Uncertainty can be indeed separated into two components, aleatoric and epistemic: the first is due to inherently random effects (such as noise in data collection) and is irreducible; the latter is caused by a lack of knowledge and can, in principle, be reduced with additional information [63]. Uncertainty quantification involves enabling an ML model to acknowledge its limitations and applying techniques that calibrate the margin of error in its predictions [134].
As a matter of fact, by providing the end users with the instruments to determine

whether they can trust the predictions made by a model or if they need extra caution while making decisions based on these predictions [101], UQ is indeed one of the most important ways to reliability achievement [102]. Modern ML systems present several sources of uncertainty, at different stages of their pipeline (pre-training, training, and post-training), which makes the reliability assessment really challenging [33].
Several UQ techniques have been studied in literature so far [2, 55], most of them focusing on reliability assurance aspects, but marginally considering its interaction with explainability.

1.3 Explainable AI and its links to reliability

The field of explainable AI (XAI) gained a lot of attention in the last few years, with a boosting of new research works and methodologies addressing the challenging topic of providing interpretable yet performant machine learning outcomes. XAI is indeed a fundamental building-block of trustworthy AI, with the objective of making the logic involved in AI-driven decisions more understandable to users, being these AI experts (e.g., developers), field experts (e.g., holders of the domain knowledge in a specific field of application of the AI system), or general users. XAI has a strong link with the *transparency* principle of trustworthy AI [60], which is also required by with many other regulations introduced worldwide so far, from the "*right of explanation*" by the European General Data Protection Regulation (GDPR) in 2018 [113], the United States Blueprint for an AI Bill of Rights [152], up to the most recent European Union AI Act [42]. The role of XAI in the latter is however still subject of debate, since, according to the Act, transparency is to be achieved through the provision of proper documentation and information to the user (e.g. instructions of use), rather than relying on XAI tools [105]. What is sure, XAI will play a central role in future AI research, and will be strictly intertwinned with the concepts of AI assurance, trustworthiness, and auditability [84].

The lack of interpretability is indeed an important issue in ML reliability assurance [101]. The higher is the level of model complexity, i.e. the more intricated is the structure of the model and the larger it is in terms of number of parameters, the higher are the limitations encountered in assessing its reliability [125]. Due to the non-linearities of black-box models, it is difficult to govern and prevent possible anomalous behaviors, such as adversarial perturbations [30]. Understanding the inner logic of a model is required to improve the knowledge about the system and facilitate the approach humans have with it, as well as to comply with regulation [7]. Also, interpretability plays a key role in discovering potential vulnerabilities and threats affecting critical infrastructures and helps increasing their overall reliability [123]. Explainability is recognized as one of the dimensions of safety assurance, and

has to coexist with other properties like robustness and performance [71]. Interpretability and uncertainty estimation both contribute to trust calibration: the first provides humans with access to what the AI system has learned, and how it uses that knowledge in producing outputs; at the same time, understanding what the AI does not know, through uncertainty estimation, is also extremely important for creating a suitable mental model of the AI capabilities [137]. Traditionally, a trade-off has been widely recognized between interpretability and accuracy, since more complex and highly performing models are usually less explainable and viceversa. However, recent research on interpretable ML argues that such a trade-off does not necessarily hold anymore, thanks to the availability of interpretable models that offer good performance and can be applied to high-stakes decision-making systems too [118]. For this reason, current XAI research is striking to design simpler models that balance interpretability and accuracy, with techniques to compensate the lack of performance caused by the XAI components [72].

1.4 PhD contribution

In the rapidly evolving context of Trustworthy AI, where the advancement of technologies and learning methods has to cope with regulation, my PhD focused on studying how to integrate reliability in explainable AI methods, thus balancing interpretability and performance.
In light of the large variety of XAI techniques, that would make it difficult to design reliability approaches in a general way for all of them, I decided to focus my research on one of the simplest forms of XAI: the rule-based models, being characterized by decision rules of the *if-then* format.
As for every data-driven learning approach, these models may often reveal sensitive to the complexity of the data under analysis, thus undermining the possibility of effectively interpreting their outcomes, even if expressed by rules. Learnt rulesets can indeed be too large in number or length, and their interpretability may not be so straightforward. Therefore, one research line that I pursued at the beginning of my PhD is the design of innovative metrics for *rule similarity*, to synthesize the syntactic and/or semantic information expressed within (and among) sets of rules, allowing to make quantitative comparisons and extract new knowledge from them. After these initial studies on rule-based models and related tools, my research started to investigate how to address reliability assessment for this category of models, which is the core of this thesis. The objective can be summarized by the following research questions:

How can explainability intersect with reliability?
How can error guarantees be provided to rule-based models?

In the big picture of the possible definitions of *reliable AI* described in Sec. 1.2,

in my thesis this concept is therefore intended as the possibility of ensuring a controlled performance, in terms of statistical error, of machine learning models. The considered setting, throughout all my PhD work, is that of binary classification for tabular data, where one class - the target class - can be typically identified as a situation in which it is desirable to minimize the prediction error of the model (e.g., predicting that patients are healthy when they are sick in reality, predicting the presence a cyber-attack when there is not any, etc.). Based on this, my contributions aim at individuating *rule-based safety regions*, i.e., regions of the feature space that ensure a reliable performance of the rule-based model on a target class. The term *safety* here recalls the practical implications that such regions have in real-world safety-critical applications, i.e., finding the conditions where the AI system keeps operating with controlled error, reducing the risk of harmful consequences for the involved people and the environment. All this while keeping the *rule-based* structure that, together with rule analysis tools such as rule similarity, can help shedding light into the conditions that provide those performance guarantees.

With this objective, I approached the problem in two different ways throughout the PhD:

- In the first year of the PhD, I devised a *heuristic* approach, where rule-based safety regions have been designed starting from the rules themselves and their properties of feature and value ranking, combining them and optimizing their thresholds so to achieve the desired error guarantees. Specifically, I designed three rule optimization methods, *reliability from inside, reliability from outside and rules with zero error*, that look at identifying the safety regions by exploiting the properties of the rules for the target class, the non-target class, and a combination of target class rules specifically trained with zero error constraint, respectively.

- In the second and third year, I investigated the same problem by relying on a solid statistical framework widely used in machine learning uncertainty quantification: *conformal prediction* (CP) [142, 12], a post-training ML validation method that outputs set predictions with proved error guarantees. I first designed a new score function - called CONFormal Interpretable-by-Design Explainable and Reliable Artificial Intelligence score (CONFIDERAI) - that allows to apply CP theory to rule-based binary classifiers, accounting for both the geometrical structure and predictive performance of decision rules. Leveraging on the results provided through CONFIDERAI, I also explored how they can be used to optimize the original rule-based model, i.e., how to derive new rules with improved error on the target class.

Due to the interest in the topics of explainability and reliability of AI for many fields of industry 4.0, throughout the three years, my research also explored the adoption of rule-based classification and of the reliable AI techniques for some applications

of interest, including adversarial machine learning, synthetic data generation, and simulated social robotics.

1.5 Thesis organization

After introducing in this Chapter 1 the main definitions in the current AI landscape within which my doctoral research was situated, the rest of the thesis is structured as follows:

- Chapter 2 provides the fundamentals and some state of the art on *eXplainable AI* (XAI) topic, with particular focus on providing the mathematical notation and definitions for rule-based classification models. It also describes the specific models adopted throughout the thesis.
- Chapter 3, after introducing some background on rule similarity, described the three novel metrics I designed: *syntactic rule similarity* (Section 3.2); *Bag Of Words similarity* (Section 3.3); *geometrical rule similarity* (Section 3.4).
- Chapter 4 describes my key PhD contributions on *rule-based safety regions*: Section 4.1 is devoted to the definition and evaluation of the heuristic approaches based on rules optimization; Section 4.2 describes my research on conformal prediction for binary rule-based classification, presenting several results on both toy and real datasets.
- Chapter 5 describes and reports the results of the methods I developed in three applications of interest in Industry 4.0: in Section 5.1, the methods from Section 4.1 are used in the context of reliable adversarial machine learning attacks detection; Section 5.2 shows the application of syntactic rule similarity from Section 3.2 for an explainable evaluation of synthetic data generation processes; finally, Section 5.3 shows how data-driven approaches, where rule-based classification is combined with uncertainty quantification techniques, can constitute a less conservative yet reliable solution with respect to model-based approaches, in the context of simulated social robotics navigation.
- Chapter 6 concludes the dissertation by providing final remarks and discussing future work.

Other activities. Even though not exactly connected to my main research line, during my doctoral journey as part of CNR-IEIIT research group on Information and Systems Engineering [1], I had the pleasure of contributing to the group's activities, by taking part to research projects of national and international relevance

[1] https://www.ieiit.cnr.it/expertise/information-systems-engineering

(see A.1), and collaborating with my colleagues on other approaches to trustworthy AI. Though still related to Reliable AI, these activities are not part of the main logical thread on the connections between explainability and reliability I want to provide with this dissertation: therefore I will report the essence of these secondary contributions in Appendix A.2. Also, I had the opportunity of spending three months during my third PhD year (April-July 2024) at the German Aerospace Center (DLR) in Braunschweig (Germany), where I started a fruitful collaboration on rule-based classifiers as a validation tool for object detection algorithm. Since this work is still in a preliminary investigation phase, and has my direct contribution in term of XAI but not (yet) in terms of Reliable AI, it is described Appendix A.3.
Finally, Appendix B outlines the journal and conference publications achieved during the PhD.

Chapter 2

Explainable Artificial Intelligence

XAI allows to enter the logic of an AI-based decision-making system by providing representations of the implicit functioning of an AI model in a human-understandable way. This Chapter provides a general overview of XAI categorization in literature, with a specific focus on rule-based models (Section 2.1). It then enters into the mathematical definitions for native ***rule-based classifiers***, also providing descriptions of specific models adopted throughout my research experiments (Section 2.2). Finally, it briefly describes local rule-based explanation methods for black-box classifiers (Section 2.3).

2.1 Overview

Despite its widespread adoption, there is not yet a full agreement among researchers on some XAI definitions [54, 84], especially around the concepts of interpretability and transparency. Therefore, several taxonomies of XAI techniques have been proposed recently with the aim of unifying the terminology to pursue clarity in the definition of regulations for ethical and reliable AI development (e.g., [53], [132]). There are three main directions to categorize XAI techniques:

1. *Transparent-by-design* (or *interpretable* or *ante-hoc explainability*) models [118], being directly designed to make predictions in an intelligible way, or *post-hoc* explainability techniques, where a black box model is first trained and a XAI technique aims to find an explanation for its outcomes [73]. Examples of techniques in the first group are rule-based models, better described in next Section 2.2, linear models, generalized linear models (GLM), generalized additive models (GAM) or Bayesian models. Well-established algorithms for post-hoc explainability include Shapley values (SHAP) [85], Local Interpretable Model-agnostic Explanations (LIME) [115] and counterfactual explanations [143].

2. *Local* explainability, which focuses on explaining the reasons behind single predictions on individual data instances, such as Anchors [116], or *global*

explainability, which in turn addresses the explanation of the whole logic of a model, as it is for decision trees or decision rules models [4].

3. *Model-agnostic* or *model-specific* techniques [16], the first providing their results independently from the underlying machine learning model (e.g., LIME or Anchors), while the latter being designed for specific models, thus including all transparent models and methods such as TreeSHAP [38].

Another way to look at XAI categorization involves considering the different data types [54] the autonomous decision-making system is dealing with, hence mainly distinguishing between tabular data, images and text.
Regarding *tabular data*, which is the data type more often used in the PhD research, widespread choices are as follows: feature importance methods, such as permutation feature importance and SHAP, that allow to identify which features contribute most to the model's predictions; partial dependence plots (PDP) [92], illustrating the relationship between a feature and the predicted outcome averaged over the dataset; local post-hoc methods such as LIME [115], to provide local explanations by approximating the black-box model with a simpler, interpretable, one through perturbation around a specific instance. However, tabular data are also well suited to inherently transparent global XAI models, such as rule-based ones. Before delving into the notation and more formal details about the algorithms adopted in this thesis, the following paragraph provides an overview of existing literature works on rule-based machine learning.

2.1.1 Rule-based models

Rule-based models refer to that transparent-by-design XAI category consisting of the most easily understandable and interpretable form of explanation for humans. The most common rules are in the form of propositional *if-then* rules, where the if clause is a combination of conditions on the input variables, and the consequent is related to the outcome (see 2.2 for more details). Many different rule learning algorithms have been proposed in this field of research, broadly divided into either rulesets of *unordered rules* or decision lists with *ordered rules* [119]. The distinction between the two mainly refers to the way they handle possible conflicts, i.e, the case of multiple rules being valid for the same input but predicting different outcomes, or the non-exhaustiveness, i.e., the presence of inputs not being covered by any rule. Imposing an ordering among rules, decision lists will use the prediction of the first rule, according to that order, satisfied by the sample. On the other hand, for unordered rulesets all the rules are tried, and the final outcome depends, typically, on some aggregation strategy such as majority voting [46]. In both ordered and unordered rules cases, the uncovered examples issues is solved through the usage of a default rule [92].

The origin of rule-based ML dates back to the end of 1980s/1990s, with pioneering works on rule learning algorithms such as, among others, CN2 [31] or RIPPER [32], both being inspired to a *separate-and-conquer* approach called covering algorithm, consisting in: 1) learning a rule that covers a part of the given training examples, 2) removing the covered examples from the training set (the separate part), and 3) recursively learn another rule that covers some of the remaining examples (the conquer part) until some stopping criterion, preventing overfitting, is met. With the boost of XAI demand, the topic of rule learning got attention in the last 10 years too. In [149], a Bayesian framework is introduced to learn a low number of short rules, seeking a balance between accuracy and interpretability through user-adjustable Bayesian prior parameters. [117] proposed EXPLORE algorithm to induce disjunctive decision rules (rule sets) in a systematic and efficient manner, based on a branch-and-bound approach that relies on user-defined performance constraints and seeks rule optimal length. In [150] generalized linear models using rule-based features are considered for regression and probabilistic classification; this solution trades off rule set complexity, in terms of number and length of rules, and prediction accuracy. Multi-value Rule Set (MRS) provides a more concise and feature-efficient model form for classification and explanation [146]. Always seeking a balance between accuracy and interpretability, authors in [78] study interpretable decision sets, i.e. sets of independent if-then rules.The learning process is based on an objective function that optimizes accuracy and interpretability of the rules using smooth local search. Rule lists are also widely investigated, being collections of ordered rules [119]. Falling rule lists [145] have been introduced, providing both predictive modelling and a descending ranking of rule outcome probability (i.e., rules leading to higher class probabilities are reported first). [10] presents an innovative technique for rule lists generation over categorical feature space, called CORELS, that gives the optimal solution according to the training objective, along with a certificate of optimality. Also, in [80] Bayesian rule lists method is investigated, combining decision lists with Bayes approaches, with applications in healthcare contexts. Rule-based models have been studied in hybrid approaches too, where they are combined to black-box models and can substitute the black-box decisions for given subsets of data [147]. Companion rule lists [104] are based on a rule list combined with a pre-trained black-box model, where users are allowed to switch between rules and the black-box, based on their requirements for interpretability or accuracy. In the context of causal inference, [148] introduces an innovative method for learning causal rule sets, based on simulated annealing. In this work, a crucial point of rule-based models is stressed out: according to the complexity (noise) of available classes of data, the number and size of the generated rules can vary a lot (i.e., complex boundaries may lead to high numbers of small rules and viceversa).

2.2 Native rule-based classification models

Native rule-based classification models are transparent-by-design machine learning models expressed as *rulesets*, i.e., sets of decision rules. Each rule is characterized by the following syntax [92]:

$$if \ \ premise \ \ then \ \ consequence$$

The *premise* part, also known as the *antecedent* of the rule, is a conjunct of conditions on the input features associated to an output class expressed by the *consequence*. It is then clear the white-box nature of these models. In the following, a formal notation for binary rule-based classifiers is provided.

Let us consider an input dataset $\mathcal{T} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{N} \in \mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^D \times \{0,1\}$, with D-dimensional feature vectors $\boldsymbol{x}_j$ and labels $y_j \in \{0,1\}$. Now, let us suppose that the feature space $\mathcal{X}$ is bounded, with $L_i \leq x_i \leq U_i$ for each feature $i = 1, \ldots, D$. A rule-based classifier is a function $g : \mathcal{X} \to \mathcal{Y}$ described by a ruleset $\mathcal{R} = \{r_k\}_{k=1}^{N_r}$. The premise of a rule r_k is the logical product of conditions c_{i_k}, each referred to a feature x_i, $i = 1, \ldots, D$ of a generic sample $\boldsymbol{x}$ and described by the interval $l_{i_k} \leq x_i \leq u_{i_k}$, with $l_{i_k} \geq L_i$ and $u_{i_k} \leq U_i$.

Remark 2.2.1 (Implicit and explicit rule conditions)**.** *The learning strategies at the basis of many rule-based models often do not end up with rules whose premise makes all features explicit: however, from a semantic point of view, this means that each value assumed by the variable that remain implicit is valid to make the rule verified, which corresponds to set $l_{i_k} = L_i$ and $u_{i_k} = U_i$ for these features.*

Following these considerations, the premise of a rule can be geometrically described by a hyper-rectangle $\mathcal{H}_{r_k}$ in the feature space, defined as:

$$\mathcal{H}_{r_k} \doteq \bigwedge_{i=1}^{D} c_{i_k}, \text{ with } c_{i_k} : l_{i_k} \leq x_i \leq u_{i_k} \tag{2.1}$$

The space individuated by the premise is associated to the output class for rule r_k, which is denoted as $\hat{y}_k$ and constitutes the *consequence* of the rule. The specific rule-based model determines if the output $\hat{y}_k$ corresponds to the final label assigned to unseen inputs or not. This depends on whether the considered inputs satisfy just one of the rules (as in the case of mutually exclusive rules such as those from decision trees) or can verify multiple, possibly conflicting, ones. In the latter case, further steps are required for class label assignment. More details on this, for the models used throughout the PhD, will be provided in the next Sections.

2.2.1 Rule relevance

Whatever it is the rule learning algorithm, the predictive performance of rule r_k can be evaluated through its covering $C(r_k)$ and error $E(r_k)$, commonly known as True Positive Rate and False Positive Rate of the rule, respectively, computed as:

$$C(r_k) = \frac{TP(r_k)}{TP(r_k) + FN(r_k)} \tag{2.2}$$

$$E(r_k) = \frac{FP(r_k)}{TN(r_k) + FP(r_k)} \tag{2.3}$$

where

$$TP(r_k) = |\{(\boldsymbol{x}_j, y_j)|\ x_{ji} \in [l_{i_k}, u_{i_k}] \wedge y_j = \hat{y}_k\}|$$

$$FP(r_k) = |\{(\boldsymbol{x}_j, y_j)|\ x_{ji} \in [l_{i_k}, u_{i_k}] \wedge y_j \neq \hat{y}_k\}|$$

$$TN(r_k) = |\{(\boldsymbol{x}_j, y_j)|\ x_{ji} \notin [l_{i_k}, u_{i_k}] \wedge y_j \neq \hat{y}_k\}|$$

$$FN(r_k) = |\{(\boldsymbol{x}_j, y_j)|\ x_{ji} \notin [l_{i_k}, u_{i_k}] \wedge y_j = \hat{y}_k\}|$$

$\forall j = 1, \ldots, N\ \forall i = 1, \ldots, D$.
The combination of covering and error provides the *rule relevance*, a metric assigning to each rule a value in the [0,1] interval (the closer to 1, the better is the rule's generalization). Specifically, rule relevance $R(r_k)$ for rule r_k is calculated as:

$$R(r_k) = C(r_k) \cdot (1 - E(r_k)) \tag{2.4}$$

2.2.2 Feature and Value Ranking.

Although inherently interpretable in their syntax, the rulesets generated for complex problems may not be trivial to understand, even from field experts. For this reason, additional ways to inspect the results of rule-based classifiers are of interest. Feature ranking (or feature importance) represents a valuable method in this direction, being able to capture which variables have the largest influence in rule generation process, thus being more relevant for the classification task. Criteria for computing the feature ranking basically depend on ordering the features based on some useful measure that evaluates the contribution of each feature to the prediction. Details for specific models will then be reported in the next Sections.

2.2.3 Logic Learning Machine

Logic Learning Machine (LLM) is a global rule-based classifier, developed by Rulex [1] as an efficient implementation of Switching Neural Networks [97]. The LLM training process occurs through the following steps:

1. *Discretization and mapping to a Boolean lattice (latticization, [108])*: variables are discretized to reduce their variability, thus increasing the efficiency of the training algorithm and the accuracy of the resulting set of rules. Then, they are mapped in a suitable Boolean lattice through the inverse only-one code [96], that preserves ordering and distances. All the strings representing single features are eventually concatenated to form a large string per each sample.

2. *Shadow Clustering* (SC) [96]: starting from the binary strings representing a training set, which can be seen as a portion of a truth table, a positive Boolean function is reconstructed via SC. This technique generates a set of implicants [98], defined as logical product terms that uniquely determine groups of points associated with a given output class. Since all the implicants are generated by looking at the whole training set, resulting rules can overlap (i.e., a sample may cover multiple rules) and represent different relevant aspects of the underlying phenomenon [109].

3. *Rule generation*: transform each implicant into a product of conditions in the original feature space (i.e., a rule), eventually finding out a set of rules.

LLM class assignment. In the inference phase, class label assignment is performed by the LLM as follows. Let us denote with $\mathcal{R}_{\boldsymbol{x}}^{y}$ the set of rules verified by a generic test point $\boldsymbol{x}$ and predicting a label y, and let us $\mathcal{R}^{y}$ be the set of all rules generated for class y. Based on these quantities, a measurement expressing how likely is that $\hat{y} = y$ is true for any $y \in \mathcal{Y}$ is given by:

$$W(\boldsymbol{x}, y) = \frac{\sum_{r \in \mathcal{R}_{\boldsymbol{x}}^{y}} R(r)}{\sum_{r \in \mathcal{R}^{y}} R(r)}.$$

Then, the following ratio defines the *predict probability* of label y as:

$$\Pr\{y|\boldsymbol{x}\} = \frac{W(\boldsymbol{x}, y)}{\sum_{c \in \mathcal{Y}} W(\boldsymbol{x}, c)}, \tag{2.5}$$

from which the label assigned to sample $\boldsymbol{x}$ is found by solving [44]:

$$\hat{y} = \arg\max_{y} \Pr\{y|\boldsymbol{x}\}. \tag{2.6}$$

It appears clear how rule relevance has an important role in determining the inference results, thus being an important indicator of model performance.

LLM value ranking. For the LLM, covering and error provide the basis for feature and value ranking definitions. First, a relevance is computed for each condition c_{i_k} as:

$$R_c(c_{i_k}) = (E(r'_k) - E(r_k)) \cdot C(r_k), \tag{2.7}$$

where r'_k denotes the rule obtained by removing condition c_{i_k} from rule r_k, and being $E(r'_k) \geq E(r_k)$. Each condition c_{i_k} refers to a specific variable x_i: let T_i be the set of all the thresholds τ_{iv} present in all conditions involving x_i across all the ruleset. Using this set, the domain $\mathrm{Dom}(x_i)$ of variable x_i can thus be subdivided into adjacent intervals $I_{iv},\ v = 1, \ldots, |T_i| + 1$, where $|T_i|$ is the cardinality of set T_i, as follows:

$$\begin{aligned}\mathrm{Dom}(x_i) = (-\infty, \tau_{i1}] \cup (\tau_{i1}, \tau_{i2}] \cup \ldots \cup (\tau_{i,v-1}\tau_{iv}] \cup \ldots \cup (\tau_{i,|T_i|+1}, +\infty) = \\ = I_{i1} \cup I_{i2} \cup \ldots \cup I_{iv} \cup \ldots \cup I_{i,|T_i|+1}\end{aligned}$$

Now, within each rule r_k in the subset $\mathcal{R}^{\hat{y}}$ of rules predicting a label $\hat{y} \in \mathcal{Y}$, each condition c_{i_k} over variable x_i is satisfied by a number n_{kv} of the intervals I_{iv}. For instance, a condition of the kind $x_i \leq \tau_{i3}$ is verified by $n_{kv} = 3$ intervals I_{i1}, I_{i2} and I_{i3}. Each of these intervals is assigned a relevance quantity given by:

$$R_k^{\hat{y}}(I_{iv}) = \frac{R_c(c_{i_k})}{n_{kv}}$$

A global relevance measure $R^{\hat{y}}(I_{iv})$ for interval I_{iv} that considers all the rules $r_k \in \mathcal{R}^{\hat{y}}$ is then computed as follows:

$$R^{\hat{y}}(I_{iv}) = 1 - \prod_{r_k \in \mathcal{R}^{\hat{y}}} \left(1 - R_k^{\hat{y}}(I_{iv})\right) \tag{2.8}$$

The *value ranking* is then defined by the descending order of these values for all possible intervals I_{iv}, thus helping individuating the intervals with the larger influence on the model for each feature.

LLM feature ranking. A relevance for the whole feature x_i can be retrieved by looking at the variation of $R^{\hat{y}}(I_{iv})$ over all the adjacent intervals I_{iv}:

$$R_x^{\hat{y}}(x_i) = (|T_i| + 1)\, \sigma_i \left(R^{\hat{y}}(I_{iv})\right) \tag{2.9}$$

where σ_i is the standard deviation of the intervals relevances. When $R^{\hat{y}}(I_{iv})$ does not change significantly across the intervals, then the thresholds defining them define parts of the input domain where the behavior of the classifier is essentially the same, and thus variable x_i has little discriminant power among different classes. The decreasing ordering of $R_x^{\hat{y}}(x_i)$ defines the *feature ranking*. An absolute feature ranking is also obtained by averaging the results of Equation 2.9 on all available labels $\hat{y} \in \mathcal{Y}$.

2.2.4 Decision Tree

Decision trees (DT) are tree-based classifiers, based on the *divide-and-conquer* paradigm, that is partitioning the feature space in an iterative way, by selecting the best feature to split the data according to some statistical metric, such as information gain, gain ratio, Gini index or misclassification error. The structure of a DT is a graph where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. Decision nodes are used to make a decision and have multiple branches, whereas leaf nodes are the output of those decisions and do not contain any further branches. By navigating from leaf to root, it's possible to identify simple intelligible rules [107]. In DT, the choice of a variable, used to separate the classes through a proper threshold, is done node-by-node. Each step (node) of the tree construction depends on the choices (variables and thresholds) made previously, during the building of the upper part of the tree (till to that node). The *divide-and-conquer* approach subsequently adds conditions to the tree based on smaller and smaller subsets of training data, thus lowering the computational time. On the other hand, it has the side effect of reducing the information available when selecting conditions after the first one. As a consequence, the rules in the resulting models are disjoint. For this reason, DT may be over-sensitive to highly relevant attributes and imbalanced datasets [47], may depend too much on the training set or may be corrupted by irrelevant attributes and noise [94].

2.2.5 Skope-Rules

Skope-rules is a rule learning algorithm aiming at building logical, interpretable and diversified rules for "scoping" a target class of interest, i.e. detecting samples from this class with high precision. Differently from the LLM or decision trees, a separate training is thus required for each class of the problem (if one wants to obtain rules for all classes).
The beginning of the learning process is inspired to that of RuleFit [45], where rules are extracted from an ensemble of trees and a weighted combination of these rules is then built by solving a L1-regularized optimization problem over the weights. The whole skope-rules process then undergoes the following phases:

1. *Bagging estimator training*: rules generation is done from a set of decision trees and/or regressors. Each path or sub-path of a branch of a tree is transformed into a decision rule. Trees are trained to predict the output class of interest. This ensures that the splits are made in such a way as to guarantee that they are meant for the prediction task.

2. *Performance filtering*: The set of rules extracted from the bagging undergoes a performance filtering based on precision and recall thresholds.

3. *Semantic deduplication*: this phase aims at selecting rules so to avoid many redundacies in their semantic content; similarity among terms (i.e., feature name and comparison operator) is thus computed based on the frequency of common terms.

In practical applications, skope-rules model finds application in several safety-critical classification tasks, as well as anomaly detection problems or cluster description [66].

2.3 Local Post-hoc Methods

2.3.1 Anchors explanations

Anchors [116] is a model-agnostic local rule extraction technique aiming at generating high-precision rules to explain point predictions of any black-box classifier. Even if they are designed to be locally faithful, these rules also hold in a certain neighborhood (perturbation space, unseen during training) of the instance being explained, thus allowing to define a measure of covering as per Eq. 2.2.
Formally, an anchor A is defined as a set of predicates on input instance x to be explained, such that:

$$\Pr\{Prec(A) \geq \lambda_{prec}\} \geq 1 - \delta, \tag{2.10}$$

where $\delta \in [0,1]$, and $\lambda_{prec} \in [0,1]$ is a threshold on precision $Prec(A)$, which is computed as:

$$Prec(A) = \mathbb{E}_{D_x(z|A)}[\mathbb{1}_{f(x)=f(z)}] \tag{2.11}$$

with f being the underlying black-box model and $D_x(z|A)$ an arbitrary distribution of perturbations z of point x when the anchor applies. Anchors are generated via a process that combines a perturbation-based mechanism with reinforcement learning. This involves candidate generation, best candidates identification via multi-armed bandit and candidate precision validation. These three steps are governed by a beam search algorithm, which is a graph search method that refines the candidate selection phase by finding those that score the highest-covering anchors[92]. Mathematically, such search of the optimal anchors can be expressed as the following combinatorial optimization problem:

$$\max_{A \text{ s.t. } \Pr\{Prec(A) \geq \lambda_{prec}\} \geq 1-\delta} C(A), \tag{2.12}$$

where $C(A)$ expresses the covering for the candidate anchor A.

Chapter 3

Novel Rule Similarity Metrics

Even if described by *if-then* expressions, rule-based classifiers may be not so trivially interpretable, especially when the inherent classification problem itself is complex. In these cases, additional tools are then required to properly interpret, and convey, the informational content of rules. This Chapter introduces three novel ***rule similarity*** metrics that allow quantitative comparisons and knowledge extraction about rule-based classifiers. It is organized as follows: Section 3.1 provides an overview of the rule similarity concept; Section 3.2 describes the ***syntactic*** rule similarity, a metric that looks at the syntax of the conditions explicitly appearing in the rules' premise; Section 3.3 presents the ***Bag of Words*** similarity, which is inspired to a text processing technique frequently used in Natural Language Processing domain, the Bag of Words, hence looking at rules as text sentences; finally, Section 3.4 defines the ***geometrical*** rule similarity, which treats rules from the geometrical perspectives by finding how they relate in the feature space.

3.1 Introduction to rule similarity

Rule similarity consists in defining a quantitative metric able to express how much two rules are similar either in their syntactic or semantic content. In the first case, rule similarity is defined based on the information contained into the rules, just treating them as textual sentence, like a human can read them (i.e., two rules are similar when they focus on closer conditions on the same input features subspace); the latter approach, on the other hand, also accounts for the data instances covered by rules instead (i.e., claiming that two rules are similar when they both cover a same large portion of samples) [61].
Generally speaking, for whatever design choice, rule similarity can be formalized as a mapping of the kind:

$$q : \mathcal{R} \times \mathcal{R} \to \mathbb{R}, \tag{3.1}$$

where $\mathcal{R}$ is a generic ruleset and it holds that:

$$q(r_k, r_z) = q(r_z, r_k) \; \forall (r_k, r_z) \in \mathcal{R}.$$

Remark 3.1.1 (Scales of values for rule similarity)**.** *In principle, rule similarity can assume any value in* $\mathbb{R}$*, with larger values associated to a higher 'closeness' of the rules being compared. However, to facilitate the interpretation, its values are often defined in a closed range, e.g., [0, 1]. In this way, given two rules* r_k *and* r_z *from* $\mathcal{R} \times \mathcal{R}$*, it is* $q(r_k, r_z) = 1$ *if* r_k *and* r_z *are exactly equal, while* $q(r_k, r_z) = 0$ *encodes completely different rules.*

Even if defined for single couples of rules, rule similarity is also useful to compare two different sets of rules: for example, when a rule-based model is trained on different datasets (of course related to a same problem, i.e., characterized by the same input variables), rule similarity between the respective rulesets may individuate how much the two datasets share the same information; furthermore, it allows to compare the knowledge expressed by two rulesets resulting from two different kinds of rule-based models. The way to achieve such comparisons simply consists in performing aggregations, such as the averaging, of $q(r_k, r_z)$ values through all the available couples $(r_k, r_z) \in \mathcal{R} \times \mathcal{R}$. Although its importance, the literature about rule similarity just reports a few methods, for example based on common similarity metrics such as the Jaccard's coefficient [129] or considering fuzzy rules [155, 49]. Throughout the PhD, novel ways for rule similarity computing were studied, leading to three different - yet common in their goal - new techniques: 1) a *syntactic* rule similarity, developed in collaboration with Rulex LLM developers (Sec. 3.2); 2) *Bag-Of-Words similarity*, based on treating rules as textual sentences (Sec. 3.3), and 3) *geometrical rule similarity*, looking at geometrical overlaps between rules in the feature space (Sec. 3.4).

3.2 Syntactic rule similarity

This rule similarity metric has been designed for the LLM model thanks to a collaboration with Rulex Innovation Labs. In particular, it is at the basis of a XAI-based framework for the evaluation of synthetic data that will be presented in Section 5.2.

Given a ruleset $\mathcal{R}$ generated by the LLM, let us consider two rules r_k and r_z drawn from it. Also, in this phase, the ruleset is assumed to only express explicit features (Remark 2.2.1). The computation of syntactic rule similarity starts by the similarity between two conditions, c_{i_k} from rule r_k and c_{j_z} from rule r_z. Condition c_{i_k} is associated to a weight

$$w_{i_k} \doteq E(r'_k) - E(r_k)$$

proportional to the increase in the rule error due to the removal of that condition. Let $\mathcal{D}_{i_k}$ be the domain associated to the condition in the feature space, corresponding to the interval $[l_{i_k}, u_{i_k}]$ for ordinal variables, and denote its size with $|\mathcal{D}_{i_k}|$ (i.e., the Euclidean distance $|u_{i_k} - l_{i_k}|$). Similarly, a condition c_{j_z} in rule r_z has a weight w_{j_z} and defines a domain $\mathcal{D}_{j_z}$ of size $|\mathcal{D}_{j_z}|$.
Considering feature x_i expressed by condition c_{i_k}, the binary quantity $\beta(\cdot)$ encodes if feature x_j covered by condition c_{j_z} is the same of c_{i_k} or not, namely:

$$\beta(c_{i_k}, c_{j_z}) = \begin{cases} 1 & \text{if } x_i = x_j \\ 0 & \text{if } x_i \neq x_j \end{cases} \tag{3.2}$$

The similarity between two conditions then depends both on the overlapping of their domains and β value, as follows:

$$q_c(c_{i_k}, c_{j_z}) = \beta(c_{i_k}, c_{j_z}) \cdot \frac{|\mathcal{D}_{i_k} \cap \mathcal{D}_{j_z}|}{\max(|\mathcal{D}_{i_k}|, |\mathcal{D}_{j_z}|)} \tag{3.3}$$

being $q_c(c_{i_k}, c_{j_z}) = 0$ if the attributes are not the same or the domains do not overlap; if the conditions are identical, it will be $q_c(c_{i_k}, c_{j_z}) = 1$.
Based on this, the similarity between two rules is computed as a normalized weighted sum of conditions similarities:

$$q_{\text{synt}}(r_k, r_z) = \frac{\sum_{i=1}^{d} \sum_{j=1}^{n} q_c(c_{i_k}, c_{j_z})(w_{i_k} + w_{j_z})}{\sum_{i=1}^{d} w_{i_k} + \sum_{j=1}^{n} w_{j_z}} \tag{3.4}$$

Such equation provides the value $q_{\text{synt}}(r_k, r_z) \in [0,1]$, being $q_{\text{synt}}(r_k, r_z) = 0$ if the rules do not contain any couple of conditions so that $q_c(c_{i_k}, c_{j_z}) > 0$ and a non-zero weight, $q_c(r_k, r_z) < 1$ if there is at least a different (or an additional) condition and $q_c(r_k, r_z) = 1$ if all the conditions in the two rules are identical.

3.2.1 Example

This Section aims at providing a step-by-step illustrative example on how rule similarity (Section 3.2) is computed.
Consider the following two sample rules:

- r_k: **if** Age $< 50 \wedge$ Income $>$ 30K $\wedge$ Experience > 10 **then** *true*
- r_z: **if** Age $> 40 \wedge$ Income $<$ 60K $\wedge$ Experience > 5 **then** *false*

involving three numerical variables: Age $\in$ [18,95]; Income $\in [10K, 150K]$ and Experience $\in [0, 40]$.
Table 3.1 reports the necessary terms to compute the numerator of Equation 3.4, assuming that weights are assigned the values reported in w_{i_k} and w_{j_z} columns,

where $i_k = \{1_k, 2_k, 3_k\}$ and $j_z = \{1_z, 2_z, 3_z\}$, since both r_k and r_z have three conditions. Equation 3.3 is applied to compute the similarity of conditions c_{i_k} and c_{j_z}: when these conditions involve different attributes (e.g., Age and Income in the second row of the Table), then their contribute to rule similarity is 0, since $\beta(c_{i_k}, c_{j_z}) = 0$ (Equation 3.2). Otherwise, the overlaps between the domains individuated by the couples of conditions can be computed and constitute the values of $q_c(c_{i_k}, c_{j_z})$ columns, which are as closer to 1 as larger is the entity of that overlap (see, e.g., the comparison between Experience > 10 and Experience > 5, which results in 0.857 similarity).

Table 3.1: Simple example showing the steps for computing syntactic rule similarity between two sample rules.

c_{i_k}	w_{i_k}	c_{j_z}	w_{j_z}	$q_c(c_{i_k}, c_{j_z})$	$q_c(c_{i_k}, c_{j_z}) \cdot (w_{i_k} + w_{j_z})$
Age < 50	0.3	Age > 40	0.2	$1 \cdot \frac{50-40}{\max(50-18,95-40)} = 0.18$	$0.18 \cdot (0.3 + 0.2) = 0.09$
Age < 50	0.3	Income < 60K	0.25	0	0
Age < 50	0.3	Experience > 5	0.1	0	0
Income > 30K	0.2	Age > 40	0.2	0	0
Income > 30K	0.2	Income < 60K	0.25	$1 \cdot \frac{60K-30K}{\max(150K-30K,60K-10K)} = 0.25$	$0.25 \cdot (0.2 + 0.25) = 0.1125$
Income > 30K	0.2	Experience > 5	0.1	0	0
Experience > 10	0.15	Age > 40	0.2	0	0
Experience > 10	0.15	Income < 60K	0.25	0	0
Experience > 10	0.15	Experience > 5	0.1	$1 \cdot \frac{40-10}{\max(40-10,40-5)} = 0.857$	$0.857 \cdot (0.15 + 0.1) = 0.215$

Substituting the numeric values in (3.4), the syntactic rule similarity between r_k and r_z is:

$$q_{\text{synt}}(r_k, r_z) = \frac{\sum_{i_k=1}^{d_k} \sum_{j_z=1}^{n_z} q_c(c_{i_k}, c_{j_z})(w_{i_k} + w_{j_z})}{\sum_{i_k=1}^{d_k} w_{i_k} + \sum_{j_z=1}^{n_z} w_{j_z}} =$$

$$= \frac{0.09 + 0.1125 + 0.215}{(0.3 + 0.2 + 0.15) + (0.2 + 0.25 + 0.1)} = 0.348$$

3.2.2 Similarity between rulesets

For practical applications (as in the scope of the work described in Sec. 5.2), it may be often useful to compare two rulesets in terms of rule similarity, by aggregating the values for different couples of rules, as detailed in the following.

Considering two different rulesets $\mathcal{R}_1$ and $\mathcal{R}_2$, the goal is to obtain a single value of rule similarity to compare them. Let us denote with $\mathcal{R}_1^y$ and $\mathcal{R}_2^y$ the subsets of the rulesets $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively, referring to the output class $y \in \mathcal{Y}$.
First, for each $y \in \mathcal{Y}$, by applying Equation 3.4, rule similarities between all possible couples of rules in $\mathcal{R}_1^y \times \mathcal{R}_2^y$ are computed.Then, the average per-class similarities between the rulesets is found as:

$$\bar{q}^y_{\text{synt}}(\mathcal{R}_1, \mathcal{R}_2) = \frac{1}{|\mathcal{R}_1^y \times \mathcal{R}_2^y|} \cdot \sum_{(r_k, r_z) \in \mathcal{R}_1^y \times \mathcal{R}_2^y} q_{\text{synt}}(r_k, r_z)$$

Eventually, the global ruleset similarity between $\mathcal{R}_1$ and $\mathcal{R}_2$ is calculated by aggregating all the class labels, as follows:

$$\bar{q}_{\text{synt}}(\mathcal{R}_1, \mathcal{R}_2) = \frac{1}{|\mathcal{Y}|} \cdot \sum_{y \in \mathcal{Y}} \bar{q}^y_{\text{synt}}(\mathcal{R}_1, \mathcal{R}_2) \tag{3.5}$$

where $\bar{q}_{\text{synt}}(\mathcal{R}_1, \mathcal{R}_2) \in [0,1]$.

3.3 Bag of words similarity

Bag-of-Words similarity [99] is an approach to rule similarity that finds its origin in the *bag-of-words (BoW)* concept, which is a common technique in Natural Language Processing (NLP)to compare texts and finding similarities. In particular, the BoW model looks at sentences as collections of words, regardless their order, but focusing just on their frequency of occurrence [110].
The starting point in BoW similarity is then to treat rules (of the *if-then* form) as special sentences, where the the *words* are derived by rule conditions. In this case, again, features not explicitly appearing in the ruleset are disregarded.
Starting from the definition in Section 2.2, each condition c_{i_k} of rule r_k (beloging to a ruleset $\mathcal{R}$) can be split into two components, as:

$$c_{i_k}: \quad x_i \geq l_{i_k} \ \wedge \ x_i \leq u_{i_k}$$

In this way, each condition is converted to a conjunct of half-open intervals, from which it is easy to identify the following two terms:

- f_{i_k} indicating the feature name corresponding to x_i variable and the associated comparison operator;
- t_{i_k} expressing the numerical threshold value for f_{i_k}.

Both these kinds of terms constitute the "*words*" of the rules. Overall, a rule r_k with N_k conditions can thus be seen as the collection

$$r_k = (\{(f_{i_k}, t_{i_k})\}_{i_k=1_k}^{N_k}, \hat{y}_k)$$

In the case of text classification, each word making the BoW is also assigned a *weight*, which is typically related to the frequency of occurrence of that word in the text, which is then thoroughly represented by a words-weight matrix called the *BoW matrix*. This same approach is thus adapted to the case of rules, by defining two matrices that can be extracted from the entire ruleset $\mathcal{R}$ to represent it in vectorial form.
The first one, denoted with F, contains information about the presence of feature-operators terms. Specifically, after removing redundant feature-operator terms in the set of words (which can occur if multiple rules present a condition with the same feature and operator), elements of matrix F are assigned according to the following criterion:

$$F(k, i_k) = \begin{cases} 1, & \text{if } f_{i_k} \in r_k \\ 0, & \text{if } f_{i_k} \notin r_k, \end{cases} \tag{3.6}$$

$\forall k = 1, \ldots, N_r$ and $\forall i_k = 1_k, \ldots, N_k$.
The second matrix, denoted with T, collects the numerical thresholds associated to each feature-operator, as follows:

$$T(k, i_k) = \begin{cases} \sigma(t_{i_k}), & \text{if } \max(t_{i_k}) \neq \min(t_{i_k}) \\ 1, & \text{if } \max(t_{i_k}) = \min(t_{i_k}) \\ 0, & \text{if } f_{i_k} \notin r_k, \end{cases} \tag{3.7}$$

where $\sigma(t_{i_k})$ is the result of the normalization of threshold values across multiple occurrences of the same feature-operator terms, given by:

$$\sigma(t_{i_k}) = \frac{t_{i_k} - \min(t_{i_k})}{\max(t_{i_k}) - \min(t_{i_k})}. \tag{3.8}$$

$\forall k = 1, \ldots, N_r$ and $\forall i_k = 1_k, \ldots, N_k$.
Now that both words and weights have been defined for rules, the BoW matrix can be built on top of them, being the matrix of size $N_r \times 2 \cdot N_k$, obtained by horizontal concatenation of F and T matrices:

$$\text{BoW} = [F|T]$$

which is a matrix representation of the ruleset $\mathcal{R}$, associating each rule r_k to a vector $\boldsymbol{v}_k$ corresponding to the k-th row of the BoW matrix.
The problem of finding similarity between rules can now be studied as computing the similarity of two vectors. Since the rules generated by a rule-based model may often be at some extent diversified one another, by design (such as the semantic deduplication in skope-rules 2.2.5) or following the training process, i.e. each of them may capture (slightly) different clusters of data, it is quite unlikely that many of them share exactly the same conditions, which generally results in a sparse BoW

matrix. This aspect should be considered when choosing a suitable metric for vector similarity. Traditional similarity measures do not work well for sparse matrices. In fact, two rules may have many 0 values in common, but this does not make them similar. Therefore, there is need of a measure that focuses on the terms that the two rules have in common and the similarity of their thresholds, thus ignoring null terms with many 0 values occurrences.
Cosine similarity is very common in text analysis, being a measure that works very well with very large sparse matrices and provides a measure of similarity just as described above. For this reason, it is adopted for the BoW rule similarity.
Considering a couple of generic rules r_k and r_z, for which corresponding vector representations $\boldsymbol{v}_k$ and $\boldsymbol{v}_z$ are found through the BoW matrix, cosine similarity is applied to compute the *Bag of Words similarity* as:

$$q_{\text{BoW}}(r_k, r_z) = \frac{\boldsymbol{v}_k \boldsymbol{v}_z^T}{\|\boldsymbol{v}_k\| \, \|\boldsymbol{v}_z\|} \tag{3.9}$$

where $\|\cdot\|$ is the Euclidean norm. When the cosine of the angle formed by the two vectors is 0, then they are perpendicular and therefore maximally different, leading to $q_{\text{BoW}}(r_k, r_z) = 0$. Conversely, as the two vectors get closer, the angle between them tends to 0 and the cosine tends to 1, progressively increasing the BoW similarity. When both vectors are aligned in the same direction, their cosine is exactly 1, bring to the maximal BoW similarity $q_{\text{BoW}}(r_k, r_z) = 1$.

3.3.1 Application example

Vehicle platooning dataset

The classification problem of collision detection in vehicle platooning [94] was adopted to experiment with the BoW similarity. In smart mobility scenarios, vehicle platooning is a challenging problem aiming at getting a group of vehicles to travel autonomously by finding a compromise between performance (e.g., speed) and safety. The fundamental idea of the platooning is that a leader vehicle controls his own speed and that of the rear vehicles by exchanging wireless messages (i.e., data packets) through a communication channel.
In the dataset considered here, all vehicles (consisting of a variable number of vehicles) are supposed to be traveling at a constant speed and at a constant reciprocal inter-vehicular distance according to the Cooperative Adaptive Cruise Control (CACC) scenario described in detail in [94]. The event that can lead to a collision is a braking force applied by the platoon leader.
The platooning dataset, generated by the Plexe simulator[128], consists of 20000 samples and includes 5 features, namely: the number of vehicles in the platoon (N), the braking force (F_0), the Packet Error Rate (PER) during communication, the initial distance between vehicles ($d0$), and the initial speed ($v0$). Based on

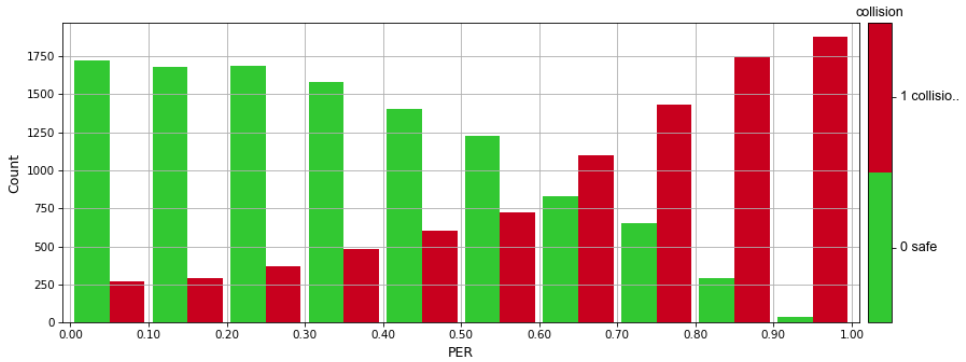


Figure 3.1: Packet Error Rate (PER) histogram of the original dataset.

these features, the classification problem consists in recognizing either *collision* or *non-collision*, where a collision is assumed to occur if two vehicles are less than 2m apart.

Scenario of interest definition

The PER variable is constant for each simulation run and is sampled by following a random uniform distribution in the interval [0.1, 0.9]. As it can be seen in the previous work [94], this variable largely impacts the occurrence of collision events. As shown in the PER histogram of Fig. 3.1, there is no a clear evidence of a PER threshold separating collision and no collision samples. How these differences affect the generation of the respective rules is then a suitable case to be studied through BoW rule similarity analysis.
To this aim, two sub-datasets have been created by splitting on PER attribute: i) PER High for $PER \geq 0.5$ (PERH in the following); ii) PER Low for $PER < 0.5$ (PERL in the following).These datasets are comparable in size (10127 and 9873 samples, respectively), but PERH registers much more collisions (6946) than PERL (1941). The LLM model (2.2.3) was trained on each of the two, leading to two sets of rules, for a total number of 39 collision and non-collision rules

Results

The BoW matrix and subsequent rule similarity are built by considering all 39 rules from PERL and PERH, with a particular interest in comparing PERL versus PERH rules.
As an example of reduced complexity, to illustrate how the BoW matrix is built, let us consider the following five rules:

r_1 (PERH): **if** $PER > 0.815 \wedge v0 > 17$ **then** *collision*
r_2 (PERH): **if** $N > 3 \wedge PER > 0.765 \wedge v0 > 27$ **then** *collision*
r_3 (PERH): **if** $N > 5 \wedge PER > 0.685 \wedge v0 > 28$ **then** *collision*

r_4 (PERL): **if** $N > 8 \wedge PER \leq 0.365 \wedge v0 > 36$ **then** *collision*
r_5 (PERL): **if** $N > 8 \wedge v0 > 42$ **then** *collision*

which are associated to the set of terms:

$$F = \{N >, PER >, PER \leq, v0 >\}.$$

Numerical thresholds associated to these terms are transformed according to Eq. 3.7, to get the values reported at the $V_{(\cdot)}$ columns in Table 3.2, which is an example of BoW matrix for the considered selection of rules.

Table 3.2: Representation of the BoW matrix of a reduced complexity example

Rules	$PER >$	$V_{PER>}$	$v0 >$	$V_{v0>}$	$N >$	$V_{N>}$	$PER \leq$	$V_{PER\leq}$
r_1	1	1	1	0	0	0	0	0
r_2	1	0.6	1	0.4	1	0	0	0
r_3	1	0	1	0.44	1	0.4	0	0
r_4	0	0	1	0.76	1	1	1	0.28
r_5	0	0	1	1	1	1	0	0

After computing the cosine similarity (Eq. 3.9) between each pair of rules in the two rulesets (PERL and PERH), values are collected in a symmetrical matrix of size 39×39. Different kinds of analyses are then performed. Similarities between rules of the two subdatasets are compared, and only 4 out of 741 PERH-PERL couples reached high levels of similarity (above 95%), as reported in Table 3.3, which would confirm the difference between the two datasets. Specifically, the involved rules are the following:

r_6 (PERH): **if** $N > 7 \wedge PER > 0.645$ **then** *collision*
r_7 (PERH): **if** $N > 8$ **then** *collision*
r_8 (PERH): **if** $N \leq 8 \wedge PER \leq 0.605$ **then** *non collision*
r_9 (PERL): **if** $N > 9$ **then** *collision*
r_{10} (PERL): **if** $N \leq 8 \wedge PER \leq 0.385$ **then** *non collision*
r_{11} (PERL): **if** $N \leq 7 \wedge PER \leq 0.425$ **then** *non collision*

High rule similarity seems related to the number of vehicles N, and partially, to PER itself.
Another example of analysis focuses on single rulesets instead, where rule similarity can be used to evidence differences between classes. Considering PERL dataset, whose rules are detailed below, it is immediately observable, as expected, that collision and non-collision rules are, in general, dissimilar and, in many cases, their similarity is even zero (see Table 3.4).

Table 3.3: BoW similarity between rules belonging to different rulesets

	r_6	r_7	r_8	r_9	r_{10}	r_{11}
r_6	1	-	-	-	-	-
r_7	0.97	1	-	-	-	-
r_8	0	0	1	-	-	-
r_9	**0.95**	**0.99**	0	1	-	-
r_{10}	0	0	**0.99**	0	1	-
r_{11}	0	0	**0.99**	0	0.99	1

r_4 (PERL): **if** $N > 8 \wedge PER \leq 0.365 \wedge v0 > 36$ **then** *collision*
r_5 (PERL): **if** $N > 8 \wedge v0 > 42$ **then** *collision*
r_9 (PERL): **if** $N > 9$ **then** *collision*
r_{10} (PERL): **if** $N \leq 8 \wedge PER \leq 0.385$ **then** *non collision*
r_{11} (PERL): **if** $N \leq 7 \wedge PER \leq 0.425$ **then** *non collision*
r_{12} (PERL): **if** $N > 8 \wedge v0 \leq 36$ **then** *collision*
r_{13} (PERL): **if** $N \leq 8 \wedge v0 \leq 57$ **then** *non collision*
r_{14} (PERL): **if** $N \leq 5$ **then** *non collision*
r_{15} (PERL): **if** $N \leq 8 \wedge F0 > -5$ **then** *non collision*
r_{16} (PERL): **if** $N \leq 9 \wedge v0 \leq 32$ **then** *non collision*
r_{17} (PERL): **if** $N \leq 9 \wedge F0 > -8 \wedge d0 > 7.485 \wedge PER \leq 0.475 \wedge v0 \leq 78$ **then** *non collision*
r_{18} (PERL): **if** $N \leq 9 \wedge PER \leq 0.145 \wedge v0 > 32$ **then** *non collision*

In addition, the similarity analysis can be devoted to a single ruleset and a single class label, and any analogies found through similarity may help pruning the ruleset to a set of significant rules, avoiding redundancies. An example of this kind follows for the collision rules of the PERH ruleset, whose similarity values are reported in Table 3.5:

r_2 (PERH): **if** $N > 3 \wedge PER > 0.765 \wedge v0 > 27$ **then** *collision*
r_3 (PERH): **if** $N > 5 \wedge PER > 0.685 \wedge v0 > 28$ **then** *collision*
r_6 (PERH): **if** $N > 7 \wedge PER > 0.645$ **then** *collision*
r_7 (PERH): **if** $N > 8$ **then** *collision*
r_{19} (PERH): **if** $N > 3 \wedge F0 \leq -2 \wedge PER > 0.605 \wedge v0 > 51$ **then** *collision*
r_{20} (PERH): **if** $N > 3 \wedge d0 > 5.3455 \wedge PER > 0.725 \wedge v0 > 28$ **then** *collision*
r_{21} (PERH): **if** $N > 3 \wedge F0 \leq -2 \wedge PER > 0.575 \wedge v0 > 63$ **then** *collision*

The syntactic knowledge derived from BoW similarity, together with covering information, can guide the expert to choose between very similar rules (e.g., r_{19} and

Table 3.4: BoW similarity of collisions versus non collision rules extracted from the PERL dataset

		PERL - collision			
		r_4	r_5	r_9	r_{12}
PERL - non collision	r_{10}	0.33	0	0	0
	r_{11}	0.34	0	0	0
	r_{13}	0	0	0	0.39
	r_{14}	0	0	0	0
	r_{15}	0	0	0	0
	r_{16}	0	0	0	0.36
	r_{17}	0.2	0	0	0.25
	r_{18}	0.53	0.34	0	0

Table 3.5: BoW similarity of collision rules extracted from PERH dataset

	r_2	r_{19}	r_3	r_{20}	r_6	r_7	r_{21}
r_2	1	-	-	-	-	-	-
r_{19}	0.65	1	-	-	-	-	-
r_3	**0.96**	0.68	1	-	-	-	-
r_{20}	**0.83**	0.56	**0.81**	1	-	-	-
r_6	0.61	0.39	0.71	0.51	1	-	-
r_7	0.47	0.36	0.64	0.41	**0.97**	1	-
r_{21}	0.61	0.99	0.65	0.53	0.36	0.34	1

r_{21}, or r_6 and r_7) and keep the most appropriate one, by selecting the appropriate trade-off between knowledge discovery and model accuracy.
Moreover, the analysis of these similarities may lead the expert to make additional considerations that may result in a growth of knowledge for the specific scenario. For example, r_2 and r_3 have very similar conditions as they all have the same terms and very close thresholds. It can be noticed, looking at Tab. 3.5, that rule r_{20}

is quite similar (with similarity larger than 80%) to these two rules. The main difference is that this rule adds information about the initial distance d_0. This new information may add a novel link that somehow did not previously emerged.
The obtained high values of rule similarity may also lead to a ruleset pruning: as an example, by removing rule r_2, which is highly similar to r_3 and r_{20}, from PERH dataset, the overall classification error increase is below 1%. Hence, rule similarity can be also considered as a way to simplify complex and less-intuitive rulesets.

3.4 Geometric rule similarity

In Section 2.2, it was described how a rule r_k can be viewed as a geometrical shape, i.e., the hyper-rectangle $\mathcal{H}_{r_k}$, that accounts for all the dimensionalities of the data at hand. The volume $\mathcal{V}(\mathcal{H}_{r_k})$ of this hyper-rectangle can be found as:

$$\mathcal{V}_{\mathcal{H}_{r_k}} = \prod_{i=1}^{D} |u_{i_k} - l_{i_k}|. \tag{3.10}$$

Hyper-rectangles are at the basis of a new *geometrical rule similarity* metric. Considering a ruleset $\mathcal{R}$, the objective is to define a rule similarity function S whose values are suitable for quantifying the overlap or the extent of adjacency between two rules, intended as the geometrical intersection (adjacency) between the corresponding hyper-rectangles.[1] Differently from previous methods, it is therefore based on the *semantics* of rules rather than on their syntax.

Let us denote two rules with r_k and r_z (both from the same set $\mathcal{R}$). An overlap (or adjacency) between them occurs if the following holds:

$$\max(l_{i_k}, l_{i_z}) \leq \min(u_{i_k}, u_{i_z}) \; \forall i = 1, \ldots, D \tag{3.11}$$

Assuming that Eq. 3.11 is satisfied for the rules, we can compute the volume of the hyper-rectangle formed by the intersection of the hyper-rectangles $\mathcal{H}_{r_k}$ and $\mathcal{H}_{r_z}$ as follows:

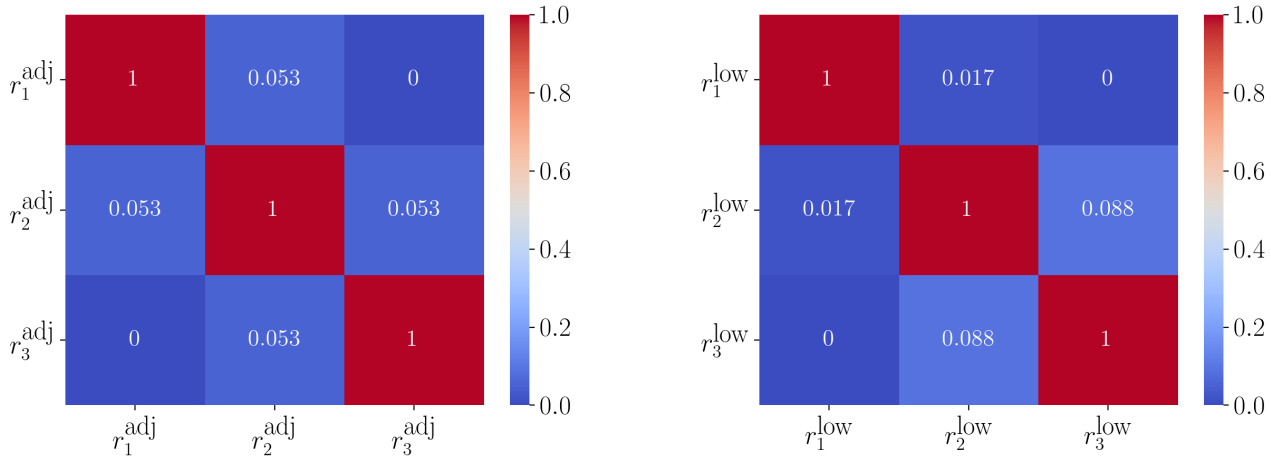
$$\mathcal{V}_{\text{overlap}(\mathcal{H}_{r_k}, \mathcal{H}_{r_z})} = \prod_{i=1}^{D} |\min(u_{i_k}, u_{i_z}) - \max(l_{i_k}, l_{i_z})| \tag{3.12}$$

Finally, Eq. 3.10 applied to r_k and r_z and Eq. 3.12 lead to the definition *geometric rule similarity*:

$$q_{\text{geom}}(r_k, r_z) \doteq \frac{\mathcal{V}_{\text{overlap}(\mathcal{H}_{r_k}, \mathcal{H}_{r_z})}}{\mathcal{V}_{\mathcal{H}_{r_k}} + \mathcal{V}_{\mathcal{H}_{r_z}} - \mathcal{V}_{\text{overlap}(\mathcal{H}_{r_k}, \mathcal{H}_{r_z})}} \tag{3.13}$$

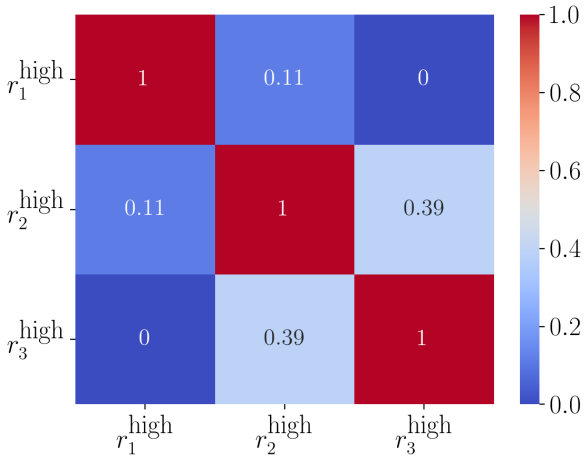
[1] In case of rule-based models that generate mutually exclusive rules (e.g, a decision trees), there are no overlaps, but still rules can be adjacent, i.e., sharing a smaller or larger surface, which one may quantify through the proposed metric.

This type of similarity has been introduced while developing the conformal prediction technique for rule-based classification. Hence, to show an applicative example of it, I refer the reader to the three sets of toy rules $\{r_1^{\text{adj}}, r_2^{\text{adj}}, r_3^{\text{adj}}\}$, $\{r_1^{\text{low}}, r_2^{\text{low}}, r_3^{\text{low}}\}$ and $\{r_1^{\text{high}}, r_2^{\text{high}}, r_3^{\text{high}}\}$ defined and illustrated in Section 4.2.2, which have been manually constructed in a simple 2D space so to have different overlap degrees.



(a) adjacency

(b) low overlap



(c) high overlap

Figure 3.2: Geometrical rule similarity values between toy rules with different entities of overlap.

Figure 3.2 shows the obtained values after Equation 3.13 has been applied to each couple of rules in the three sets. It can be observed that the obtained values in all cases correctly reflect the respective positioning of the rules in the feature space (see also Fig. 4.5): it assigns the lowest value (0) to couples of rules that do not

share any boundary, such as for $r_1^{(\cdot)}$-$r_3^{(\cdot)}$; values as larger than 0 as the extension of the overlap regions increases; the highest values (1) when the two rules perfectly coincide.

Chapter 4

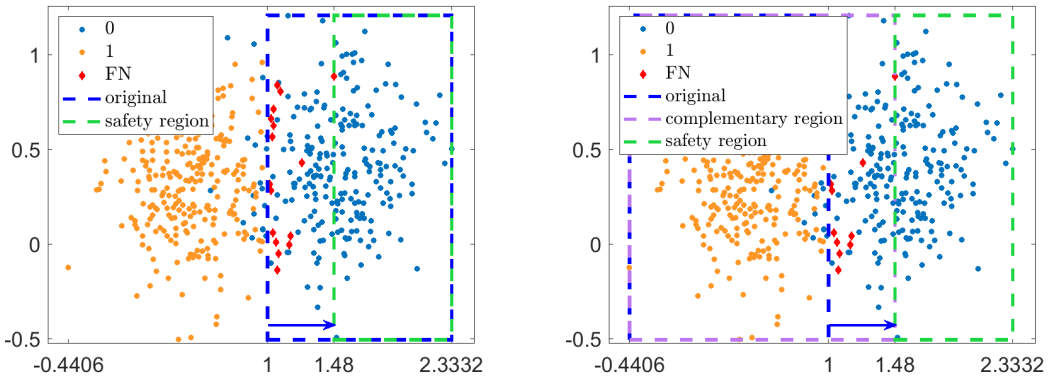
Rule-based Safety Regions

This chapter, which can be considered the core of the research conducted throughout the PhD, presents two approaches that have been studied and developed with the aim of providing reliability to rule-based classifiers, through the design of ***rule-based safety regions*** in the feature space associated to a guaranteed statistical error of the rule-based models. On the one hand, the methods presented in Section 4.1 pursue this goal by heuristically acting on the rule-based classifiers only. On the other hand, the research works illustrated in Section 4.2 follow the same final objective, but relying on an additional tool for machine learning verification, i.e., the conformal prediction.

4.1 Rules optimization for error control

The problem of designing *safety regions* in the feature space that ensure a statistical control on the error of rule-based classifiers on a target class has been first addressed, at the beginning of the PhD, through the design of a heuristic approach. It involves directly exploiting the generated rules, and their feature and value ranking properties.
Referring to a binary classification problem, suppose to label the target class with $y = 0$, over which the error control is sought. This class expresses a *safe* situation (e.g., the absence of a collision in autonomous driving, or the absence of a disease in healthcare context, etc.). In contrast, label $y = 1$ is used to denote the presence of a *critical/unsafe* situation.
Based on these definitions, the objective is to build safety regions characterized by the minimum level of false negatives (FN), that are the points of the class $y = 1$ being wrongly classified as safe by the model. These regions are designed through a process of sensitivity analysis applied to rules thresholds, filtered via feature and value ranking. Two complementary approaches have been introduced to this purpose, namely *reliability from inside* and *reliability from outside*, which share the

same algorithmic structure yet being diverse in some points.
Before going into the details and the formalization of the methods, the illustrative example of Fig. 4.1 might help explaining the intuition behind them. As described



(a) Inside concept

(b) Outside concept

Figure 4.1: Graphical 2D illustration concept behind the idea of *reliability from inside* (in short, Inside, left) and *reliability from outside* (in short, Outside, right).

in Section 2.2, a binary rule-based classifier generates a set of rules for predicting each of the classes. When looking for safety regions made up of only target points ($y = 0$), two possibilities arise: on the one hand, the focus can be posed on the rules predicting the target class itself (blue dashed rectangle in Fig.4.1a), and opportunely tune their thresholds to achieve a region that excludes any point of the other class covered by them (i.e., the FNs, red diamonds in the figures), as represented by the green dashed rectangle in Fig. 4.1a; on the other hand, the search could start from the rules predicting the non-target class (blue dashed rectangle in Fig.4.1b), then their thresholds could be tuned so that they cover all points of their same class they did not previously cover (i.e, again, the FN points), obtaining a region (purple dashed rectangle in Fig.4.1b) whose complementary gives the safety region (again, the green dashed rectangle in Fig. 4.1b). These approaches constitute the ideas behind reliability from inside and reliability from outside algorithms, respectively. As a third method for comparison, the rule-based classifier itself can be trained with statistical zero error for each rule, which is the starting point of the *rules with zero error* algorithm, presented in Section 4.1.4.

4.1.1 Algorithm common structure

Algorithm 1 RuleBasedReliability

Inputs:
Dataset $\mathcal{T} = \{(\boldsymbol{x}_k, y_k)\}_{k=1}^{D}, \boldsymbol{x}_k \in \mathbb{R}^N, y_k \in \{0,1\}$
Target class $y = 0$
N_{FR} number of most relevant features
Grid of candidate perturbations $\mathcal{H}$

1. Train a rule-based classifier on $\mathcal{T}$
2. Select N_{FR} features from feature ranking
3. Find $[s_j, t_j]$ from value ranking
4. Define hyper-rectangle $\mathcal{I} = \bigcup_{j=1}^{N_{FR}} [s_j, t_j]$
5. $\forall \Delta_i \in \mathcal{H}$, do:
   Define a new hyper-rectangle:
   $\mathcal{P}(\Delta_i) = \bigcup_{j=1}^{N_{FR}} [s_j \mp \delta_{s_j}^i \cdot s_j, t_j \pm \delta_{t_j}^i \cdot t_j] = \bigcup_{j=1}^{N_{FR}} [\tilde{s}_j, \tilde{t}_j]$
6. Find optimal perturbations Δ^*

Algorithm 1 shows the common algorithm residing behind the idea of both reliability from inside and outside methods. Since pursuing rule thresholds tuning can be computationally expensive when considering the whole set of rules generated in a real-world context, these two methods are designed exploit a selection of the most discriminative features and intervals of values, as individuated by the feature and value ranking derived from the rule-based model.

After training the rule-based classifier on the available dataset $\mathcal{T}$ (step 1), feature and value ranking are used to select the N_{FR} top-relevance intervals $[s_j, t_j]$, $j = 1, \ldots, N_{FR}$, where N_{FR} is the number of selected features, which is set by design (steps 2-3). The logical union of these intervals is then considered to shape the structure of the region (step 4). In step 5, the tuning phase begins, by applying perturbations of the kind $\boldsymbol{\delta}_j = (\delta_{s_j}, \delta_{t_j})$ to the initial thresholds of the interval related to feature $j = 1, ..., N_{FR}$. Candidate values to be explored for these perturbations are chosen so that the following constraints hold:

$$\min(\boldsymbol{x}^j) \leq s_j \mp \delta_{s_j} \cdot s_j \leq \max(\boldsymbol{x}^j)$$

$$\min(\boldsymbol{x}^j) \leq t_j \pm \delta_{t_j} \cdot t_j \leq \max(\boldsymbol{x}^j)$$

where $\boldsymbol{x}^j$ is the dataset component corresponding to feature j. These inequalities individuate ranges of acceptable values for δ_{s_j} and δ_{t_j}, denoted with $[\delta_{s_j}^{min}, \delta_{s_j}^{max}]$ and $[\delta_{t_j}^{min}, \delta_{t_j}^{max}]$, that guarantee that the candidate perturbation maintains the new thresholds within the same ranges of the training data.

Perturbation values are then sampled within such ranges by using a fixed step h_j, which generates:

$$N_{s_j} = \left\lfloor \frac{\delta_{s_j}^{max} - \delta_{s_j}^{min}}{h_j} \right\rfloor + 1$$

possible perturbations for threshold s_j, and

$$N_{t_j} = \left\lfloor \frac{\delta_{t_j}^{max} - \delta_{t_j}^{min}}{h_j} \right\rfloor + 1$$

for threshold t_j, leading to

$$N_j = N_{s_j} + N_{t_j}$$

possible threshold values for feature j. Then, by considering all the features, the total number of perturbations is given by:

$$N_c = \binom{N_j}{2},$$

that considers all combinations for the possible threshold values.
The *perturbation candidates grid* is then found as:

$$\mathcal{H} = \{\Delta_i | \Delta_i = (\boldsymbol{\delta}_1^i, ..., \boldsymbol{\delta}_{N_{FR}}^i),\ \boldsymbol{\delta}_j^i = (\delta_{s_j}^i, \delta_{t_j}^i),\ j = 1, ..., N_{FR},\ i = 1, ..., N_c\} \quad (4.1)$$

For each Δ_i, a novel hyper-rectangle $\mathcal{P}(\Delta_i)$ is defined, which is a rigid expansion or reduction of the initial hyper-rectangle $\mathcal{I}$ defined at point 3.
The optimal perturbations Δ^* providing guarantees of reduced FNs are then individuated among all the candidates in $\mathcal{H}$. The next two Sections will detail how the two methods, inside and outside, differentiate with respect to the presented structure.

4.1.2 Reliability from Inside

As previously anticipated, this method starts from the rules of the target class ($y = 0$). Therefore, in points 2 and 3 of Algorithm 1, the most influent features and corresponding intervals will be picked from feature and value rankings of the target class. These intervals are *restricted* until leaving out all false negatives, hence perturbations of the kind $[s_j + \delta_{s_j} \cdot s_j, t_j - \delta_{t_j} \cdot t_j]$ of point 5 are considered. The optimal perturbations Δ^* are then found by solving the following problem:

$$\Delta^* = \underset{\Delta_i : N_1 = 0}{\arg\max}\, \mathcal{V}(\mathcal{P}(\Delta_i)) \quad (4.2)$$

where N_1 is the number of points in $\mathcal{T}$ classified as $y = 1$ and included into the hyper-rectangle $\mathcal{P}$ with volume $\mathcal{V}$. This problem is formulated as the maximum volume to avoid trivial solutions that exclude FN points but leaving no (or 'too

few') TN points inside the region. The safety region is then represented by the region $\mathcal{P}(\Delta^*)$.
In 2D, for $N_{FR} = 2$, such region is represented by the surface:

$$\mathcal{P}(\Delta^*) = [s_1 + \delta^*_{s_1} \cdot s_1, t_1 - \delta^*_{t_1} \cdot t_1] \vee [s_2 + \delta^*_{s_2} \cdot s_2, t_2 - \delta^*_{t_2} \cdot t_2], \tag{4.3}$$

with

$$\Delta^* = \begin{bmatrix} \delta^*_{s_1} & \delta^*_{s_2} \\ \delta^*_{t_1} & \delta^*_{t_2} \end{bmatrix}$$

4.1.3 Reliability from Outside

Differing from the previous method, reliability from outside starts from the non-target class rules, i.e., $y = 1$ in this case.
In this case, points 2 and 3 of Algorithm 1 are referred to feature and value ranking of class $y = 1$. False negatives can then be reduced by *expanding* the original intervals, i.e., this time by looking at perturbations of the kind $[s_j - \delta_{s_j} \cdot s_j, t_j + \delta_{t_j} \cdot t_j]$. The solution for the optimal Δ^* is then found by solving:

$$\Delta^* = \arg \min_{\Delta_i : N_1 = D_1} \mathcal{V}(\mathcal{P}(\Delta)), \tag{4.4}$$

where D_1 is the number of points in $\mathcal{T}$ with label $y = 1$. The optimal $\mathcal{P}$, in this case, corresponds to a region containing all the non-target points (like, e.g., the purple region in 4.1b), and therefore the safety region is given by $\mathcal{P}'(\Delta^*)$, where $\mathcal{P}'$ is the complementary to $\mathcal{P}$.
This time, in a 2D setting, the safety region has the following expression:

$$\begin{aligned} \mathcal{P}' = ((-\infty, s_1 - \delta^*_{s_1} \cdot s_1) \vee (t_1 + \delta^*_{t_1} \cdot t_1, \infty)) \wedge \\ ((-\infty, s_2 - \delta^*_{s_2} \cdot s_2) \vee (t_2 + \delta^*_{t_2} \cdot t_2, \infty)). \end{aligned} \tag{4.5}$$

4.1.4 Rules with Zero Error

In some contexts, where the boundaries between classes might be very complex, the exclusive use of feature and value ranking intervals for designing safety regions may reveal not enough to capture that complexity. Therefore, another solution should be sought to handle this problem, by looking for more complex separators than a simple union of intervals, still preserving the zero FN constraint. The method presented here, called *rules with zero error*, exploits the possibility of training a rule-based classifier so that the error of each rule r_k predicting the target class is set to zero by design. Let us denote this kind of ruleset with $\mathcal{R}_{0\%}$, defined as:

$$\mathcal{R}_{0\%} = \{r_k | E(r_k) = 0,\ k = 1, \ldots, M, \hat{y}_k = 0\} \tag{4.6}$$

Generating a set of rules with zero error, however, does not often come in a short and (highly) interpretable result, but rather it is more likely that a larger number

of rules with lower covering are created in order to fulfill the error limit. It is therefore ineffective to use this ruleset in its original form, since it might not result in a sufficiently explainable safety region. To this end, another optimization, again based on feature ranking, is needed.
Within $\mathcal{R}_{0\%}$, the top M_C highest-covering rules are selected and joint through a logical OR ($\vee$) operator, thus building a new predictor $\hat{r}$. Being a union of rules, $\hat{r}$ is expected to have increased covering than single rules, but loosing the 0% error guarantees. Thus, pursuing the same goal as for reliability from inside and outside algorithms, a suitable tuning of the thresholds of $\hat{r}$ is necessary so that the error (representing the FNs of the classification problem) is again minimized. Conditions $l_j \leq x_j \leq u_j$ of the rules composing $\hat{r}$, and referred to the top N_{FR} features from the feature ranking of the target class $y = 0$, are then considered in the tuning process, while the other conditions remain unchanged. For each feature $j = 1, \ldots, N_{FR}$, perturbations of the kind $[l_j + \delta_{l_j} \cdot l_j, u_j - \delta_{u_j} \cdot u_j]$ are applied, as the conditions of interest (and, consequently, the volume underlying $\hat{r}$) should be restricted to avoid false negatives. Denoting with $\boldsymbol{\delta} = \{\boldsymbol{\delta}_j\}$, with $\boldsymbol{\delta}_j = (\delta_{l_j}, \delta_{u_j})$ the perturbation matrix for all features, and being $\hat{r}(\boldsymbol{\delta})$ be the resulting predictor, the objective is then to find the optimal $\boldsymbol{\delta}$ by solving the following problem:

$$\boldsymbol{\delta}^* = \underset{\boldsymbol{\delta}: E(\hat{r}(\boldsymbol{\delta}))=0}{\arg\max} \; C(\hat{r}(\boldsymbol{\delta})) \tag{4.7}$$

The solution gives the optimal predictor $\hat{r}(\boldsymbol{\delta}^*)$, which constitutes the safety region.

4.1.5 Experiments and results

The methods described above have been applied on two different classification problems of meaningful interest from the point of view of safety: physical fatigue detection in working task simulation and collision avoidance in vehicle platooning. The LLM model is chosen as the reference for the experiments. The performance of reliability of inside and outside methods is compared, in terms of false negative rate (FNR), which is the optimization metric, but also based on the true negative rate (TNR), which reflects the quantity of safe points covered by a safety region. The aim is to obtain the largest TNR with zero FNR.

Datasets

Physical fatigue This open-source dataset [127] collects Inertial Movement Units (IMU)-based and heart-rate monitor measurements from 15 participants while they were asked to execute a simulation of an industrial task, called Manual Material Handling, for a duration of 180 minutes. While performing the task, four IMU devices are located at the ankle, hip, wrist and chest of the participants, capturing acceleration data, from which jerk and 3D space posture measures were derived.

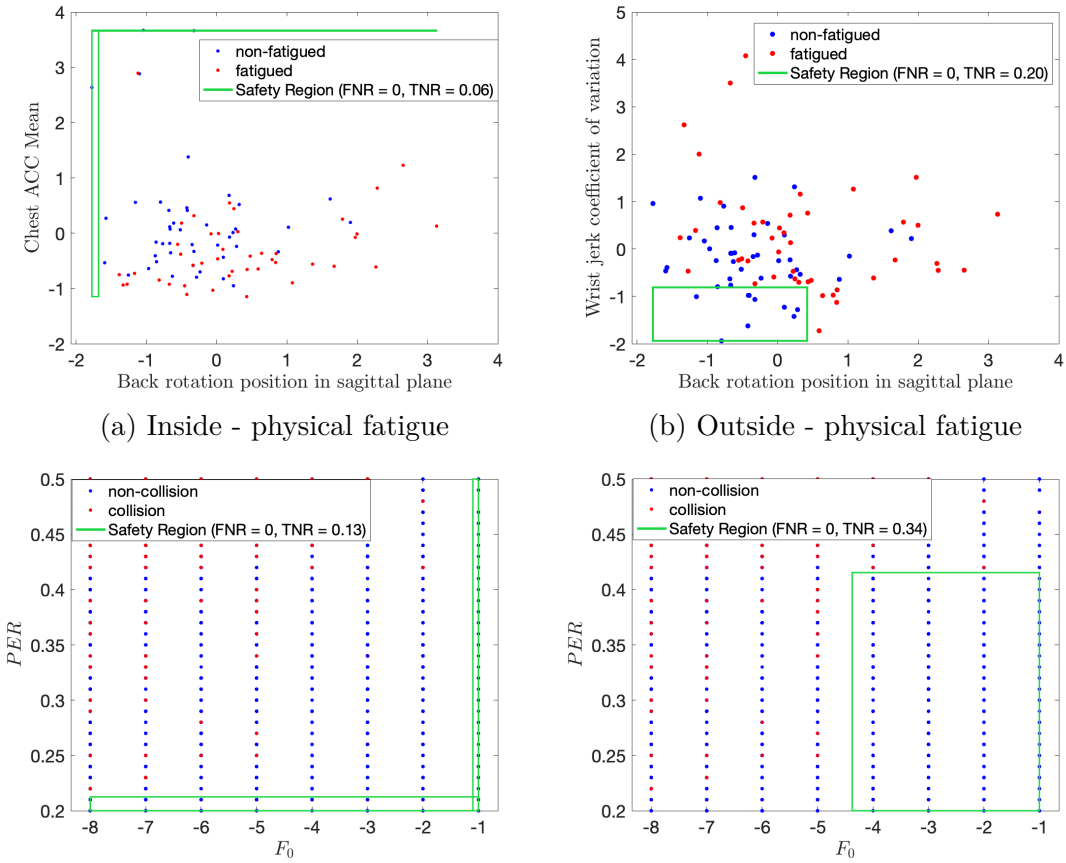
Furthermore, a heart rate monitor device was put on the chest, and the %HRR (Heart Rate Reserve) was calculated to express the percentage of an individual's heart rate being used under effort. For all these kinds of measure, mean and coefficient of variation (CV) were computed. In addition, individual (age, gender) and biomechanical features were considered. For a complete list of the original features, see Table 2 in [127]. The labelling of these measurements into *fatigued* or *non-fatigued* classes is based on self-reporting the Rate of Perceived Exertion through the Borg Scale [153] every 10 minutes, which is a widespread scale for physical fatigue assessment, assigning numbers from 6 to 20 corresponding from "None" to "Very, very hard" levels. Specifically, a RPE$\geq$13 corresponds to a fatigued state, whereas lower values constitute the non-fatigued class.
For deriving safety regions, only age, IMU-based and biomechanical features were considered, for a total number of 38 variables. The *non-fatigued* class is the target class ($y = 0$): indeed, a safety region is here intended as finding values of the feature space where the absence of fatigue is highly guaranteed.

Vehicle platooning In this application, a subset of the platooning dataset introduced in Sec. 3.3.1 is considered. Specifically, a subset of $D = 10^4$ samples is obtained after filtering the 5 features in the following ranges: 1) the number of vehicles $N \in$ [3,8]; 2) the braking force $F_0 \in [-8, -1] \times 10^3$ N; 3) the Packet Error Rate $PER \in$ [0.2,0.5]; 4) the initial distance between vehicles $d_0 \in$ [4,9] m (supposed constant for all of them); 5) the initial speed $v0 \in$ [10,90]km/h. Two classes are assigned: *collision*, when distance between two vehicles is lower than 2 m, and *non-collision* otherwise. The latter class is the target for the design of the safety regions, which then aims at finding guarantees of non-collisions.

Performance evaluation

Table 4.1: **Inside and outside.** Performance metrics (false negative rate, FNR and true negative rate, TNR) of the safety regions $\mathcal{P}(\Delta^*)$ and $\mathcal{P}'(\Delta^*)$ obtained, respectively, with reliability from inside and reliability from outside methods in the two datasets. The original region $\mathcal{I}$ (step 4 of Algorithm 1) is also reported.

		Original Region	**Safety Region**	**FNR**	**TNR**
Physical fatigue	Inside	(back rotation position in sagittal plane $\leq$ 0.03)$\vee$ (chest acceleration mean $>$ -0.47)	(back rotation position in sagittal plane $\leq$ -1.68)$\vee$ (chest acceleration mean $>$ 3.65)	0	0.06
	Outside	(back rotation position in sagittal plane $>$ 0.03) $\vee$ (wrist jerk coefficient of variation $>$ 0.03)	(back rotation position in sagittal plane $<$ 0.42) $\wedge$ (wrist jerk coefficient of variation $<$ -0.81)	0	0.2
Vehicle Platooning	Inside	PER $\leq 0.33 \vee F_0 > -3.50$	PER $\leq 0.21 \vee F_0 > -1.1$	0	0.13
	Outside	PER $> 0.43 \vee F_0 < -7.50$	PER $\leq 0.415 \wedge F_0 \geq -4.37$	0	0.34



(a) Inside - physical fatigue

(b) Outside - physical fatigue

(c) Inside - platooning

(d) Outside - platooning

Figure 4.2: Graphical representations of the 2D safety regions obtained on physical fatigue and vehicle platooning datasets with reliability from inside and reliability from outside methods.

Inside versus Outside Table 4.1 reports the metrics related to the inside and outside methods tested on the two case studies, by assuming $N_{FR} = 2$ for simplicity and computational efficiency.
On physical fatigue dataset, the value ranking individuated *back rotation position in sagittal plane* $\leq$ 0.03 and *chest acceleration mean* > -0.47 as the two most relevant intervals for predicting non-fatigue. The optimization through the inside method led to individuate the new thresholds of -1.68 and 3.65, respectively, achieving the goal of FNR=0. However, the size of the region was not as satisfactory, covering only the 6% of points of non-fatigued points.
Conversely, a better performance is shown by the outside method, where the feature ranking individuated the same first variable (*back rotation position in sagittal plane* > 0.03) , but a different second one (*wrist jerk coefficient of variation* > 0.03). In

this case, the optimal FNR = 0 is reached with a much higher TNR of 20%. Indeed, the optimal threshold for the first variable is lower (0.42) with outside than with inside method (1.68).
In the platooning dataset, the performance is overall better than for the fatigue. Once again, the outside method gives larger TNR at the optimal FNR. Interestingly, this time the feature ranking evidenced the same 2 most influent variables, i.e., PER and F_0, for both non-collision and collision classes.
The green areas over the classes scatter plots in Fig. 4.2 plots illustrate the obtained safety regions. Such visualizations confirm the more noisy nature of the physical fatigue dataset with respect to the vehicle platooning cases, which is reflected in the smaller size (measured through TNR) of the regions.

Rules with zero error As shown by the low values of TNR metrics of Table 4.1, inside and outside methods tend to individuate small (yet optimal) safety regions. This overall impacts on the efficiency of the inherent systems, requiring a too stringent choice of the variables of interest. Therefore, rules with zero error from Section 4.1.4 has also been investigated. The algorithm was applied on the two datasets by considering: i) the LLM trained with 0% maximum error per rule, and ii) skope-rules, by setting its *precision_min* parameter (i.e, the minimum precision for each rule) to 1, which is equivalent of zero error.
Concerning the LLM model, results are shown below. The blue boxes report the predictors $\hat{r}$ formed by the selection of $M_C = 4$ rules with the largest covering, before solving the optimization. The green boxes report the predictors $\hat{r}(\boldsymbol{\delta}^*)$ after the perturbation. Bold conditions denote the $N_{FR} = 2$ features chosen by feature ranking for the tuning and the respective thresholds. In this regard, the lower bound of feature *HipACCMean* (the first by importance) in the physical fatigue data experienced a large change when moving from the original (0.51) to the optimized (1.45) value, while the second feature *WristjerkMean* undergoes a small change. In the platooning case, the optimization results in a slight modification of the original zero error rules with respect to $v0$ (as the inherent $\boldsymbol{\delta}_j$ is very low) and with a significant impact on PER as it is reduced to its lower bound in the second rule.
As for the previous methods, the performance is evaluated based on FNR and TNR. As expected, it is possible to observe that, for both datasets, the TNR of the safety region is higher than both inside and outside, thanks to the involvement of many more features (especially in a higher-dimensional problem such as the physical fatigue) and the combination of multiple rules. In both experiments, the FNR lowers after the thresholds tuning, even if only a suboptimal solution was individuated for the platooning case.

Physical fatigue - LLM (original)

if (**0.51** < **HipACCMean** ≤ 1.98 ∧ ChestACCcoefficientofvariation ≤ 1.11 ∧ -1.73 < averagestepdistance ≤ 0.81 ∧ backrotationpositioninsagplane ≤ 0.52) ∨ (**WristjerkMean** > **0.55** ∧ -1.35 < Back rotation position in sag plane ≤ 0.04) ∨ (-1.73 < averagestepdistance ≤ -0.22 ∧ backrotationpositioninsagplane ≤ -0.25 ∧ -0.44 < numberofsteps ≤ 3.75 ∧ -1.73 < Wristjerkcoefficientofvariation ≤ 0.55) ∨ (ChestxpostureMean > -0.033 ∧ HipzpostureMean > 0.43 ∧ WristACCMean > -0.83 ∧ -0.88 < backrotationpositioninsagplane ≤ 0.29)
then non-fatigued
`FNR = 0.06, TNR = 0.75`

Physical fatigue - LLM0% safety region

if (**1.45** < **HipACCMean** ≤ 1.98 ∧ ChestACCcoefficientofvariation ≤ 1.11 ∧ -1.73 < averagestepdistance ≤ 0.81 ∧ backrotationpositioninsagplane ≤ 0.52) ∨ (**WristjerkMean** > **0.53** ∧ -1.35 < Back rotation position in sag plane ≤ 0.04) ∨ (-1.73 < averagestepdistance ≤ -0.22 ∧ backrotationpositioninsagplane ≤ -0.25 ∧ -0.44 < numberofsteps ≤ 3.75 ∧ -1.73 < Wristjerkcoefficientofvariation ≤ 0.55) ∨ (ChestxpostureMean > -0.033 ∧ HipzpostureMean > 0.43 ∧ WristACCMean > -0.83 ∧ -0.88 < backrotationpositioninsagplane ≤ 0.29)
then non-fatigued
`FNR = 0.00, TNR = 0.42`

Vehicle platooning - LLM (original)

if ($N \leq 5 \wedge v0 \leq 54.50$)∨
(***PER*** ≤ **0.295** ∧ $N \leq 7 \wedge v0 \leq 86.50$) ∨
(***v0*** ≤ **28.50** ∧ $PER \leq 0.445$) ∨
(***v0*** ≤ **28.50** ∧ $N \leq 6 \wedge d_0 \leq 7.86$)
then non-collision
`FNR = 0.05, TNR = 0.55`

Vehicle platooning - LLM0% safety region

if ($N \leq 5 \wedge v0 \leq 54.50$)∨
(***PER*** ≤ **0.212** ∧ $N \leq 7 \wedge v0 \leq 86.50$) ∨
(***v0*** ≤ **28.47** ∧ $PER \leq 0.445$) ∨
(***v0*** ≤ **28.47** ∧ $N \leq 6 \wedge d_0 \leq 7.86$)
then non-collision
`FNR = 0.02, TNR = 0.45`

Results obtained with skope-rules model are reported in the red and violet boxes. As before, the original rules and related performance metrics are reported (red boxes) for comparison with the optimized versions (violet boxes), and the thresholds changes are highlighted with bold font.
Concerning physical fatigue dataset, the baseline (original) rules performed worse

than those generated via LLM, being FNR = 0.11 (versus 0.06). Despite the relatively large thresholds variations involved, the perturbation process reached a sub-optimal solution, without being able to achieve a FNR lower than 0.07. Notably, this value is still larger than the LLM-derived value before optimization. Moreover, the features of interest, resulting from skope-rules feature importance, were different from those individuated by the LLM, which reflects the difference of the two models in classifying fatigue data.
In the case of vehicle platooning, the performance achieved by skope-rules is comparable with that of the LLM, both in the pre-optimization phase (0.04 FNR versus 0.05) and after (0.02 FNR with both models). However, with skope-rules, TNR value of the safety regions is only the 5% lower than the original (while it was 10% lower with the LLM); furthermore, unlike the LLM model, the optimal thresholds reach larger values, especially the speed $v0$, that moves from 45.5 in the original to 75 in the optimized rules. This indicates a larger ability of skope-rules in finding a sub-optimal safety region while maintaining a sufficiently large parameters range.

Physical fatigue - skope-rules (original)

if (**backrotationpositioninsagplane** $\leq$ **0.08** $\wedge$ HipjerkMean $>$ -1.03 $\wedge$
HipACCcoefficientofvariation $\leq$ 0.75 $\wedge$ HipypostureMean $\leq$ 1.12 $\wedge$
HipzpostureMean $>$ -1.78) $\vee$
(backrotationpositioninsagplane $\leq$ 0.17 $\wedge$ **Wristjerkcoefficientofvariation** $\leq$ **0.05** $\wedge$
HipACCMean $>$ -0.47) $\vee$
(backrotationpositioninsagplane $\leq$ 0.22 $\wedge$ Wristjerkcoefficientofvariation $\leq$ 0.06 $\wedge$
HipACCMean $>$ -0.10 $\wedge$ ChestjerkMean $>$ -1.36)
then non-fatigued
`FNR = 0.11, TNR = 0.69`

Physical fatigue - skope-rules 0% safety region

if (**backrotationpositioninsagplane** $\leq$ **-0.06** $\wedge$ HipjerkMean $>$ -1.03 $\wedge$
HipACCcoefficientofvariation $\leq$ 0.75 $\wedge$ HipypostureMean $\leq$ 1.12 $\wedge$
HipzpostureMean $>$ -1.78) $\vee$
(backrotationpositioninsagplane $\leq$ 0.17 $\wedge$ **Wristjerkcoefficientofvariation** $\leq$ **-0.74** $\wedge$
HipACCMean $>$ -0.47) $\vee$
(backrotationpositioninsagplane $\leq$ 0.22 $\wedge$ Wristjerkcoefficientofvariation $\leq$ 0.06 $\wedge$
HipACCMean $>$ -0.10 $\wedge$ ChestjerkMean $>$ -1.36)
then non-fatigued
`FNR = 0.07, TNR = 0.67`

Vehicle platooning - skope-rules (original)

if $(PER \leq 0.41 \wedge \boldsymbol{v0} \leq \mathbf{45.5}) \vee$
$(N \leq 7.5 \wedge F_0 > -7.5 \wedge \boldsymbol{PER} \leq \mathbf{0.32}) \vee$
$(N \leq 5.5 \wedge v0 \leq 54.5) \vee$
$(F_0 > -4.5 \wedge PER \leq 0.41 \wedge v0 > 64.5)$
then non-collision
`FNR = 0.04, TNR = 0.57`

Vehicle platooning - skope-rules 0% safety region

if $(PER \leq 0.41 \wedge \boldsymbol{v0} \leq \mathbf{75}) \vee$
$(N \leq 7.5 \wedge F_0 > -7.5 \wedge \boldsymbol{PER} \leq \mathbf{0.37}) \vee$
$(N \leq 5.5 \wedge v0 \leq 54.5) \vee$
$(F_0 > -4.5 \wedge PER \leq 0.41 \wedge v0 > 64.5)$
then non-collision
`FNR = 0.02, TNR = 0.52`

Discussion

By comparing the three methods presented throughout Section 4.1, the following points of discussion emerged.
The problem of ensuring the reliability of a rule-based machine learning binary classifier is something that highly depends on the structure of the data under analysis. As evidenced by the obtained results, the three algorithms performed differently in the two datasets. More specifically, the LLM 0% achieved optimality (zero FNR) for the physical fatigue problem and suboptimality (almost zero FNR) in vehicle platooning. Nevertheless, it always outperformed skope-rules with *precision_min* set to 1, which stopped at a suboptimal solution level for both datasets. As expected, in both case studies and methods, zero error rules results showed an improvement in TNR with respect to the other methods.
On the other hand, the inside-outside methods based on Algorithm 1 show flexibility in looking at the feature space, alternating good results (outside in platooning) and bad results (inside in physical fatigue). The outside approach finds larger (higher TNR) safety regions than the inside both in fatigue and platooning, highlighting that also looking at the non-target class rules (i.e., rules describing the class of safety-critical situations) when searching for safety assurance is a valuable alternative. Since the methods depend on feature and value ranking, this can involve considering the same set of features emerged for the target class (as happened in the platooning) or different ones (as partially occurred in the physical fatigue case). When the first option holds, the outside solution may be even joined with the inside, leading to a larger and more complex safety region.
As an example with the platooning case, the safety regions of figures 4.2c and 4.2d, involving PER and F_0, give rise to the area represented in Fig. 4.3.

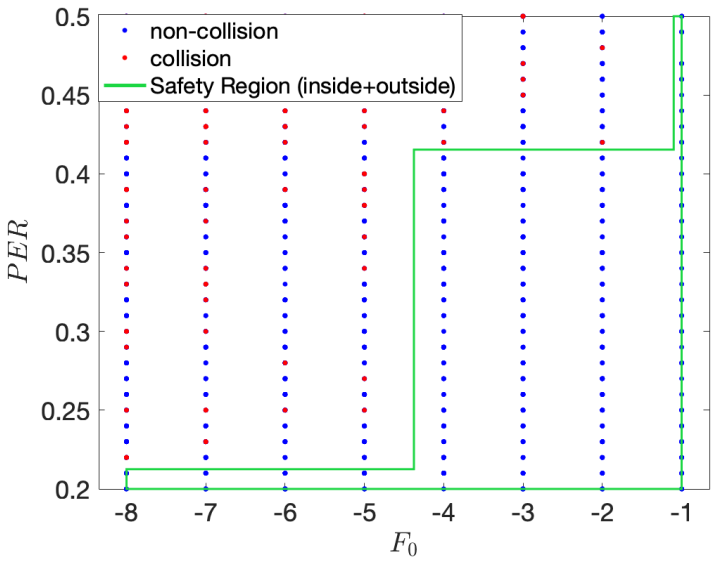


Figure 4.3: Representation of the joint region obtained by combining inside and outside solutions in the vehicle platooning case, where the feature ranking agrees on both the classes.

4.1.6 Conclusions

This Section proposed native rule-based classification models as a solution for reliability assurance. In particular, this research has shown how innovative rule optimization algorithms, that only rely on decision rules' properties, can be applied to design rule-based safety regions in the feature space with zero statistical error. The methods looked at the problem from different perspectives, offering a set of good alternatives to be selected according to the nature of the problem under analysis.

4.2 Conformal prediction for rule-based binary classifiers

In their essence, all the methods described so far drive the error control through a "grid-search"-like approach, where optimal rule thresholds to achieve the desired error target are determined heuristically by searching within sets of candidates. Even though the results on the tested cases are promising, such a search strategy might pose computational issues, especially when the optimization dimensionality N_{FR} goes above the 2D or 3D. Moreover, the achieved error level is also driven by heuristics, thus lacking of a more formal theoretical background.

To face these limitations, the same objective of defining safety regions for rule-based classifiers has been approached under the conformal prediction theory. The

research work described in this Section thus introduces CONFIDERAI (CONFormal Interpretable-by-Design score function for Explainable and Reliable Artificial Intelligence), a new score function to build conformal predictors for rule-based classifiers, 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, as well as possible overlaps among rules. Also, the concept of *conformal critical set (CCS)* [26], i.e., the set of target points for which CONFIDERAI indicates high probabilistic guarantees of the underlying model, is exploited to improve the performance of the original rules. In particular, such a set allows to redefine the classification problem at hand, by individuating a new labeling strategy that proves useful in achieving a new set of *modified rules* with improved precision with respect to the original ones, still maintaining the explainability of the final classifier. Next Sections thus first describe the fundamental concepts of CP theory, and then illustrate how CONFIDERAI score is designed, how it works in toy examples, and how the conformal critical set is built.

4.2.1 Conformal prediction theory

Calibration [141] constitutes an important post-training technique in machine learning, aimed at ensuring that the system's UQ process is accurate enough to avoid overconfidence/underconfidence issues. A strong and widespread calibration technique, also related to selective prediction, is *conformal prediction*, a statistical learning framework that allows for the construction of prediction regions with provable guarantees [12]. Conformal predictors are gaining increasing importance in ML, since they can be exploited to easily validate algorithms in terms of confidence on the prediction and efficiency in the performance. It is mainly an a-posteriori verification of the designed classifier, and in practice returns a measure of its "conformity" to the calibration data.
Within CONFIDERAI, the so-called "inductive" CP is considered: in this setting, starting from the ML model trained on a *proper training set* $\mathcal{Z}_{tr} = \mathcal{X}_{tr} \times \mathcal{Y}_{tr}$ and a held-out *calibration set*, CP allows to construct a new set predictor with proved probabilistic guarantees, which is finally evaluated on a *test set* through useful evaluation metrics.

The first key step is the definition of a *score function* $s : \mathcal{X} \times \mathcal{Y} \longrightarrow \mathbb{R}$, where $\mathcal{X}$ is a measurable feature space and $\mathcal{Y} = \{0,1\}$ a binary label space. Given a point $\boldsymbol{x} \in \mathcal{X}$ and a *candidate label* $\hat{y} \in \mathcal{Y}$, the score function returns a score $s(\boldsymbol{x}, \hat{y})$. Larger scores encode worse agreement between point $\boldsymbol{x}$ and the candidate label $\hat{y}$. Then, a calibration set of n_c observations is defined as follows:

$$\mathcal{Z}_{cal} \doteq \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c} = \mathcal{X}_{cal} \times \mathcal{Y}_{cal},$$

The observations $\boldsymbol{x}_i \in \mathcal{X}_{cal}$ are assumed to be drawn from the same probability

distribution of the observations in a test set $\mathcal{Z}_{ts} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_{ts}} = \mathcal{X}_{ts} \times \mathcal{Y}_{ts}$. Additionally, CP requires that the data are *exchangeable* [142], which is a weaker assumption than the i.i.d. hypothesis. Exchangeability means that the joint distribution of the data $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n$ is unchanged under permutations:

$$(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n) \sim (\boldsymbol{z}_{\sigma(1)}, \boldsymbol{z}_{\sigma(2)}, \ldots, \boldsymbol{z}_{\sigma(n)}), \text{ for all permutations } \sigma.$$

Then, given a user-defined error level $\varepsilon \in (0,1)$, a *prediction set* $\mathcal{C}_\varepsilon(\boldsymbol{x})$ is defined as the set of candidate labels whose score function is lower than the $(\lceil (n_c + 1)(1 - \varepsilon) \rceil / n_c)$-quantile, denoted as s_ε, computed on the $s_1, \ldots, s_{n_c}$ calibration scores, after ordering them in a non-decreasing way.
That is, to every point $\boldsymbol{x}$, CP associates a set of labels:

$$\mathcal{C}_\varepsilon(\boldsymbol{x}) = \{\hat{y} \mid s(\boldsymbol{x}, y) \leq s_\varepsilon\} \in 2^{\mathcal{Y}}, \tag{4.8}$$

The usefulness of the prediction set is that, according to [142], $\mathcal{C}_\varepsilon(\boldsymbol{x})$ is *valid*, i.e., it fulfills the so-called *marginal coverage guarantee* property:

$$1 - \varepsilon \leq \Pr\{y \in \mathcal{C}_\varepsilon(\boldsymbol{x})\} \leq 1 - \varepsilon + \frac{1}{n_c + 1}, \tag{4.9}$$

where “marginal" means that the probability is averaged over the randomness of the calibration set. In other words, the true label y belongs with high probability – at least $(1 - \varepsilon)$ – to the prediction set.

Evaluation metrics

The results of conformal prediction are typically evaluated by considering the average error and size associated to the obtained prediction sets for test points in $\mathcal{Z}_{ts}$.
The *average error* (*AvgErr*) reflects the frequency with which the true label of a point falls outside the prediction set, serving as an indicator of how often the conformal predictor "misses" the correct class. The error rate should ideally match the desired error level ε, providing that the method is well-calibrated. Formally, it is:

$$AvgErr = \frac{1}{n_{ts}} \sum_{i=1}^{n_{ts}} \mathbb{1}\{y_i \notin \mathcal{C}_\varepsilon(\boldsymbol{x}_i)\} \tag{4.10}$$

where $\mathbb{1}\{\cdot\}$ denotes the indicator function. In binary classification setting with $\mathcal{Y} = \{0, 1\}$, a separate average error for each class can be computed as:

$$AvgErr0 = \frac{1}{n_{ts}^0} \sum_{i=1}^{n_{ts}^0} \mathbb{1}\{y_i \notin \mathcal{C}_\varepsilon(\boldsymbol{x}_i) \wedge y_i = 0\} \tag{4.11}$$

and

$$AvgErr1 = \frac{1}{n_{ts}^1} \sum_{i=1}^{n_{ts}^1} \mathbb{1}\{y_i \notin \mathcal{C}_\varepsilon(\boldsymbol{x}_i) \wedge y_i = 1\} \tag{4.12}$$

being n_{ts}^0 and n_{ts}^1 the numbers of test set points with label $y_i = 0$ and $y_i = 1$, respectively.
The size of prediction sets gives an idea of how specific or broad they are. Smaller prediction sets mean more precise predictions but might come at the cost of a higher error rate, while larger sets provide higher coverage but may be less informative. Hence, measuring size reflects the *efficiency* of the conformal prediction. In the considered binary classification context, the following possible $\mathcal{C}_\varepsilon(\boldsymbol{x}_i)$ can occur:

- $\mathcal{C}_\varepsilon(\boldsymbol{x}_i) = \emptyset$, i.e., the prediction set is *empty*, with cardinality 0;
- $\mathcal{C}_\varepsilon(\boldsymbol{x}_i) = \{0\}$ or $\mathcal{C}(\boldsymbol{x}_i) = \{1\}$, i.e., the prediction set is *singleton*, with cardinality 1;
- $\mathcal{C}_\varepsilon(\boldsymbol{x}_i) = \{0, 1\}$, i.e., the prediction set is *double*, with cardinality 2.

Collecting the average occurrences of the three situations, the following metrics are defined:

$$AvgEmpty = \frac{1}{n_{ts}} \sum_{i=1}^{n_{ts}} \mathbb{1}\{|\mathcal{C}_\varepsilon(\boldsymbol{x}_i)| = 0\} \tag{4.13}$$

$$AvgSingle = \frac{1}{n_{ts}} \sum_{i=1}^{n_{ts}} \mathbb{1}\{|\mathcal{C}_\varepsilon(\boldsymbol{x}_i)| = 1\} \tag{4.14}$$

$$AvgDouble = \frac{1}{n_{ts}} \sum_{i=1}^{n_{ts}} \mathbb{1}\{|\mathcal{C}_\varepsilon(\boldsymbol{x}_i)| = 2\} \tag{4.15}$$

where $|\mathcal{C}_\varepsilon(\boldsymbol{x}_i)|$ is the cardinality of the prediction set related to test point $\boldsymbol{x}_i$. The quantity $AvgSingle$ can be also split into separate rates for the two labels, namely:

$$AvgSingle0 = \frac{1}{n_{ts}} \sum_{i=1}^{n_{ts}} \mathbb{1}\{\mathcal{C}_\varepsilon(\boldsymbol{x}_i) = \{0\}\} \tag{4.16}$$

and

$$AvgSingle1 = \frac{1}{n_{ts}} \sum_{i=1}^{n_{ts}} \mathbb{1}\{\mathcal{C}_\varepsilon(\boldsymbol{x}_i) = \{1\}\} \tag{4.17}$$

Ideally, the prediction set should have large $AvgSingle$ and low $AvgEmpty$ and $AvgDouble$.

4.2.2 A score function for rule-based classifiers

In CP, the behavior of a suitable score function $s(\boldsymbol{x}, y)$ leads to higher values for any label y that is less likely to be the correct prediction for the considered point $\boldsymbol{x}$. In this research, the focus is set on designing a new score function suitable for

applying conformal prediction theory to rule-based machine learning models with the properties described in Sec. 2.2.
Specifically, it is important to remind that the boundary of any decision rule r_k of a ruleset $\mathcal{R}$ defines a hyper-rectangle $\mathcal{H}_{r_k}$ of volume $\mathcal{V}_{\mathcal{H}_{r_k}}$ in a D-dimensional feature space, and that the geometrical juxtaposition of two rules can be measured via the geometrical rule similarity as described in Sec. 3.4. Viewing a ruleset as a set of hyper-rectangles possibly overlapping or adjacent one another is key to the design of the CONFIDERAI score function.
Given a ruleset $\mathcal{R}$ generated after training a rule-based classifier, the definition of the score function starts by considering each single rule r_k composing it, computing a score for it, and then aggregating scores for the set of rules $\mathcal{R}_{\boldsymbol{x}}$ that cover a same input $\boldsymbol{x}$.
It can be reasonably argued that points lying inside the volume of r_k, but farther from the boundary, and closer to the barycenter, are most probably well conforming to the class label y predicted by the rule; conversely, for points located closer to the rule boundary, the surroundings of the boundary need to be further examined before deciding about conformity. Indeed, two different situations may occur: i) r_k overlaps or is adjacent to other rules predicting the *same class label* y and/or ii) r_k overlaps or is adjacent to other rules predicting the *opposite class* (denoted with $\neg y$) . As a result, the score function needs penalizing (i.e., assigning larger score values) more the latter scenario, while not penalizing (i.e., assigning lower score values) the former: in this respect, geometrical rule similarity comes into play as a way to account for such overlaps.
Fig. 4.4 illustrates the overall conformal prediction pipeline, with specific focus on the components making of the CONFIDERAI score function, which will be defined step-by-step in the following.
First, let us introduce the quantity $\hat{\gamma}(\boldsymbol{x}, r_k)$, which encodes the geometrical contribution of the score function:

$$\hat{\gamma}(\boldsymbol{x}, r_k) \doteq \begin{cases} \gamma(\boldsymbol{x}, r_k) \cdot \frac{\bar{q}_{\text{geom}}(r_k, \mathcal{R}^{y}_{\boldsymbol{x}} \setminus r_k)}{\bar{q}_{\text{geom}}(r_k, \mathcal{R}^{\neg y}_{\boldsymbol{x}})} & \text{if} \quad \frac{\bar{q}_{\text{geom}}(r_k, \mathcal{R}^{y}_{\boldsymbol{x}} \setminus r_k)}{\bar{q}_{\text{geom}}(r_k, \mathcal{R}^{\neg y}_{\boldsymbol{x}})} \neq 0 \\ \gamma(\boldsymbol{x}, r_k) \qquad \text{otherwise} \end{cases} \tag{4.18}$$

It is in turn composed of two terms: the first term, $\gamma(\boldsymbol{x}, r_k)$, representing the distances of a point $\boldsymbol{x}$ from the boundary of rule r_k, is obtained as:

$$\gamma(\boldsymbol{x}, r_k) = \sum_{i=1}^{D} \left(\frac{1}{d_i^-(\boldsymbol{x}, c_{i_k})} + \frac{1}{d_i^+(\boldsymbol{x}, c_{i_k})} \right), \tag{4.19}$$

with

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

Remark 4.2.1 (Design of $\gamma(\boldsymbol{x}, r_k)$)**.** *Considering $\varphi(d) : \mathbb{R} \to \mathbb{R}$ a monotonically decreasing (scalar) function of the point-to-boundary distance d, the design choice*

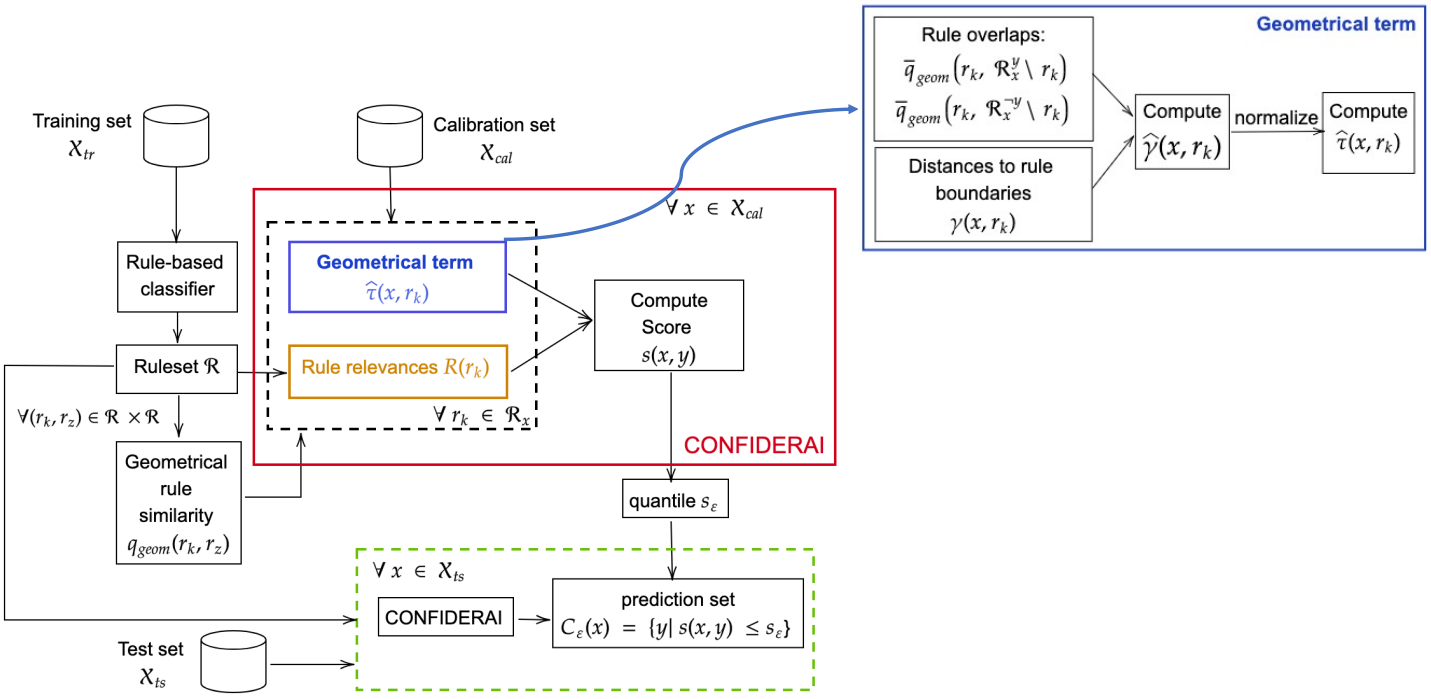


Figure 4.4: Conformal prediction for rule-based classifiers workflow, with focus on CONFIDERAI score function design (red square): combined contribution of the geometrical term (blue square) and the performance term (orange square). The geometrical term, in turn, encodes information on how a point is located within rule boundaries and on rule overlaps.

is here to let $\varphi(d) = \frac{1}{d}$*: in Eq. 4.19, the term* γ *is indeed defined as the sum of inverse distances. However, other choices might be possible, such as setting* $\varphi(d) = \exp(-\alpha d)$ *and controlling the velocity of the descent through* α *parameter.*

The second term of (4.18) handles the possible overlaps/adjacency with other rules, exploiting the concept of geometrical rule similarity as per Eq. 3.13. Specifically, the numerator $\bar{q}_{\text{geom}}(r_k, \mathcal{R}_{\boldsymbol{x}}^y \backslash r_k)$ expresses the average geometrical rule similarity between r_k and the set $\mathcal{R}_{\boldsymbol{x}}^y \backslash r_k$ of the other adjacent/overlapped rules predicting the *same* class label of r_k:

$$\bar{q}_{\text{geom}}(r_k, \mathcal{R}_{\boldsymbol{x}}^y \backslash r_k) = \frac{\sum_{r_z \in \mathcal{R}_{\boldsymbol{x}}^y \backslash r_k} q(r_k, r_z)}{\# \mathcal{R}_{\boldsymbol{x}}^y \backslash r_k} \tag{4.20}$$

Conversely, the denominator is the average rule similarity between r_k and the set

$\mathcal{R}_{\boldsymbol{x}}^{\neg y}$ of its adjacent/overlapped rules predicting the *opposite* class label:

$$\bar{q}_{\text{geom}}(r_k, \mathcal{R}_{\boldsymbol{x}}^{\neg y} \backslash r_k) = \frac{\sum_{r_z \in \mathcal{R}_{\boldsymbol{x}}^{\neg y} \backslash r_k} q(r_k, r_z)}{\# \mathcal{R}_{\boldsymbol{x}}^{\neg y} \backslash r_k} \tag{4.21}$$

Finally, to let $\hat{\gamma}(\boldsymbol{x}, r_k)$ vary in the [0,1] range, the sigmoid function is applied:

$$\hat{\tau}(\boldsymbol{x}, r_k) = \frac{1}{1 + e^{-\hat{\gamma}(\boldsymbol{x}, r_k)}} \tag{4.22}$$

The geometrical term $\hat{\tau}(\boldsymbol{x}, r_k)$ is then used in combination with rule relevance to define the *score* for point $\boldsymbol{x}$ and class label y:

$$s(\boldsymbol{x}, y) \doteq \prod_{r_k \in \mathcal{R}_{\boldsymbol{x}}^{y}} \hat{\tau}(\boldsymbol{x}, r_k)(1 - R(r_k)), \tag{4.23}$$

where the product is on the set $\mathcal{R}_{\boldsymbol{x}}^{y}$ of rules predicting label y and verified by the input point $\boldsymbol{x}$. The contribution of rule relevance 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 explain the behavior of the score function, illustrative examples are shown in the following.

Example 1: toy rules

Geometrical contribution to the score function. The toy examples presented here are devoted to illustrate how the score changes when rules with different overlap degrees are considered, thus highlighting the importance of using both the distance-based and the geometrical rule similarity-based terms. For simplicity, let us consider a bi-dimensional feature space formed by variables X_1 and X_2, on which three sets of toy rules are generated ad-hoc. The first set of rules is composed of three rules that are adjacent to each other:

$$r_1^{\text{adj}}\text{: } \textbf{if } 0.07 < X_1 \leq 0.27 \wedge 0.60 < X_2 \leq 1.00 \textbf{ then } y = 0$$

$$r_2^{\text{adj}}\text{: } \textbf{if } 0.27 < X_1 < 0.80 \wedge 0.40 < X_2 \leq 0.75 \textbf{ then } y = 1$$

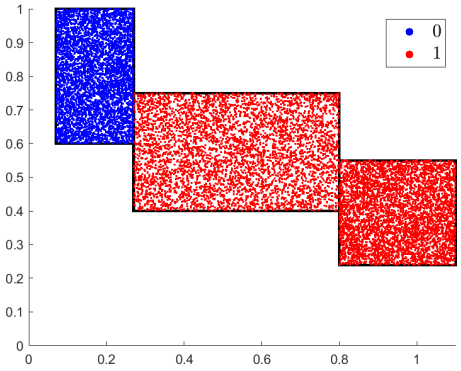
$$r_3^{\text{adj}}\text{: } \textbf{if } 0.80 < X_1 \leq 1.10 \wedge 0.24 < X_2 \leq 0.55 \textbf{ then } y = 1$$

Then, consider an intermediary case where r_1^{adj} and r_3^{adj} are slightly modified to generate a small overlap with rule r_2^{adj}, resulting in the following rules:

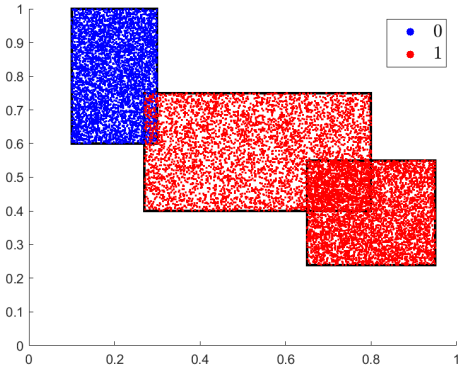
$$r_1^{\text{low}}\text{: } \textbf{if } 0.10 < X_1 \leq 0.30 \wedge 0.60 < X_2 \leq 1.00 \textbf{ then } y = 0$$

$$r_2^{\text{low}}\text{: } \textbf{if } 0.27 < X_1 < 0.80 \wedge 0.40 < X_2 \leq 0.75 \textbf{ then } y = 1$$

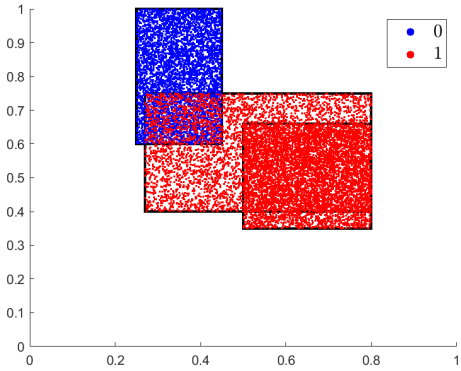
$$r_3^{\text{low}}\text{: } \textbf{if } 0.65 < X_1 \leq 0.95 \wedge 0.24 < X_2 \leq 0.55 \textbf{ then } y = 1$$



(a) adjacency



(b) low overlap



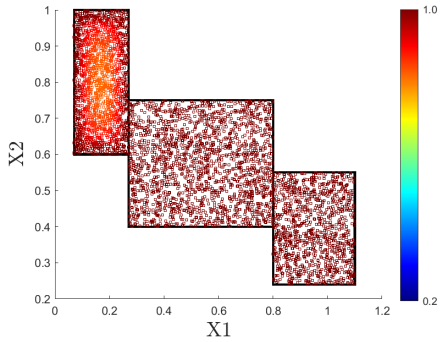
(c) high overlap

Figure 4.5: Scatter plots of the toy rulesets.

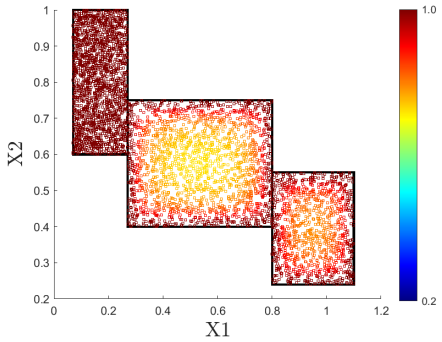
Lastly, more pronounced overlaps are considered in the third set:

$$r_1^{\text{high}}\text{: } \textbf{if } 0.10 < X_1 \leq 0.30 \wedge 0.60 < X_2 \leq 1.00 \textbf{ then } y = 0$$
$$r_2^{\text{high}}\text{: } \textbf{if } 0.27 < X_1 < 0.80 \wedge 0.40 < X_2 \leq 0.75 \textbf{ then } y = 1$$
$$r_3^{\text{high}}\text{: } \textbf{if } 0.50 < X_1 \leq 0.80 \wedge 0.35 < X_2 \leq 0.66 \textbf{ then } y = 1$$

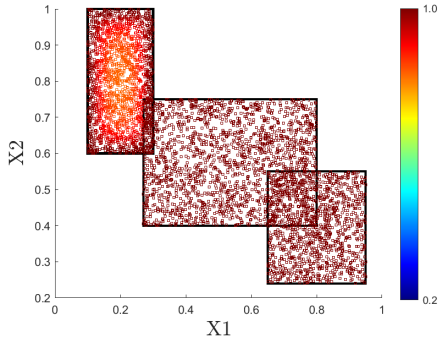
Figure 4.5 shows the scatter plots of points from the two classes within the corresponding rule boundaries.
These toy rules are useful to understand the behavior of the designed score function. In these examples, rule relevances are assumed to be 0, in order to only show the geometrical contribution of the score. Figure 4.6, from top to bottom, shows the variation of the score as the overlap between the rules increases. In each plot, rule $r_1^{(\cdot)}$ is the leftmost along the X_1 axis, $r_2^{(\cdot)}$ is at the middle, while rules $r_3^{(\cdot)}$ is the rightmost. The left column of the Figure refers to class $y = 0$ as the label for



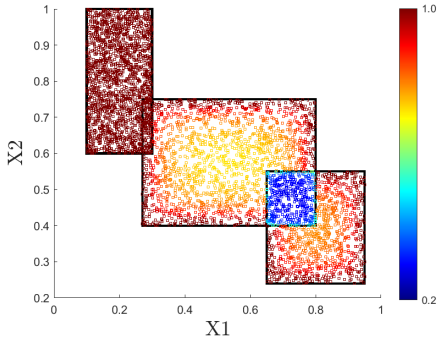
(a) adjacency - $y = 0$



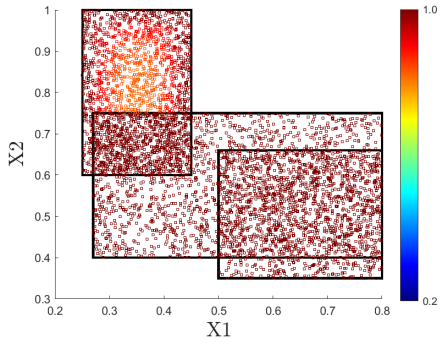
(b) adjacency - $y = 1$



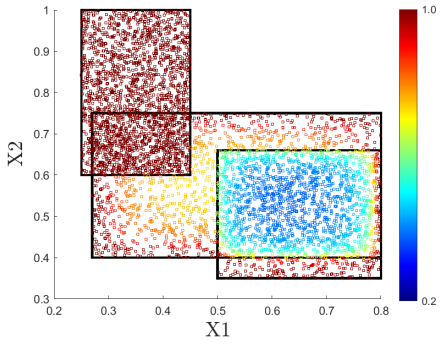
(c) low overlap - $y = 0$



(d) low overlap - $y = 1$



(e) high overlap - $y = 0$



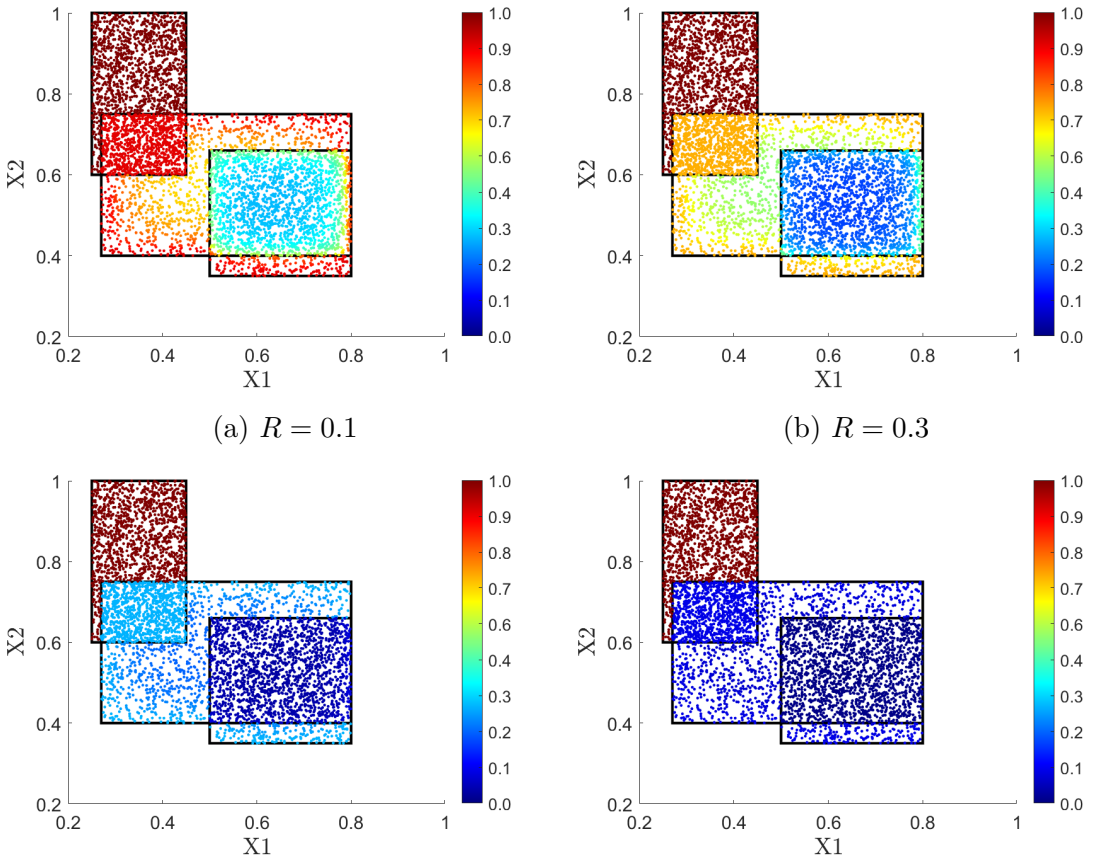
(f) high overlap - $y = 1$

Figure 4.6: Changes in the values of the score $s(\boldsymbol{x}, y)$ for points within a set of toy rules designed to have different overlap levels. Rule relevance is assumed 0, thus showing the geometrical contribution of the score.

computing $s(\boldsymbol{x}, y)$, and the right column relates to class $y = 1$. A first general observation is that, as expected, the values of the score for $r_1^{(\cdot)}$ are lower when computed for label $y = 0$ than for $y = 1$ (where they score 1). Similarly, values for $r_2^{(\cdot)}$ and $r_3^{(\cdot)}$ are lower for label $y = 1$ and fixed to 1 for $y = 0$. By looking at the

plots row-wise, it is possible to see that the score function: i) generates high scores for points lying within the overlap area between rules that predict *different* outputs (see, e.g., the overlap between r_1^{high} and r_2^{high} in Fig. 4.6e, where the score is 1); ii) generates lower scores for points enclosed in the overlap of rules that predict the *same* output labels (see, e.g., the intersection rectangle between r_2^{high} and r_3^{high} in Fig. 4.6f).

Rule relevance impact. As already anticipated, rule relevance acts as a "modulation" factor that tunes the geometrical term of the score function according to the predictive ability of the rules. The following example is thus devoted to illustrate the effects of such a modulation, showing how different values of rule relevance impact on the score values.



(a) $R = 0.1$ (b) $R = 0.3$

(c) $R = 0.7$ (d) $R = 0.9$

Figure 4.7: Example showing the effect of increasing relevance value on the score function for a set of toy rules; the higher the relavance, the lower is the score for the correct label ($y = 1$).

Considering the same third toy ruleset $\{r_1^{\text{high}}, r_2^{\text{high}}, r_3^{\text{high}}\}$ introduced in the previous example above (Fig.4.5c), increasing values of rule relevances are manually set, namely:

$$R(r_1^{\text{high}}) = R(r_2^{\text{high}}) = \quad R(r_3^{\text{high}}) = \{0.1, 0.3, 0.7, 0.9\}$$

For simplification purposes, the same relevance is here assumed for all rules.
The trend of the score values for the chosen relevances is shown in Fig. 4.7, which only refers to the score computed with label $y = 1$, since analogous considerations would hold for label $y = 0$ too. By looking at the plots, it is possible to observe that, when the relevance value is low (e.g., 0.1 or 0.3), the score values mainly depend on the geometrical contribution defined by Equations (4.18) and (4.22): indeed, points that are closer to rule boundaries are more distinguishable than the others. Conversely, as the relevance grows (see $R = 0.7$, Fig. 4.7c), its contribution

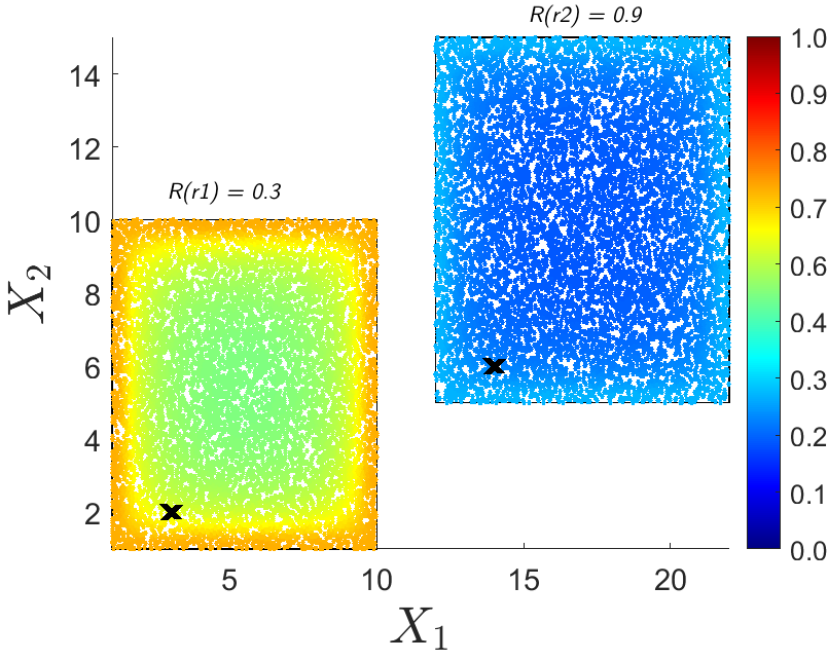


Figure 4.8: Toy example showing two toy rules r_k, $k = \{1,2\}$ with relevance $R(r_1) = 0.3$ and $R(r_2) = 0.9$, respectively, whose boundaries share the same aspect ratio. The black cross point in r_1 has a higher score than the one in r_2.

gets more significant, by lowering the score value even for points that lie close to the boundaries. This is even more pronounced in the case of the highest relevance ($R = 0.9$, Fig. 4.7d), where the predictive ability of the rule is so high to overwhelm the geometrical contribution.
In practice, this behavior of the score handles the possible case when multiple rules have the same geometrical shape (in terms of the aspect ratio of their boundary), but different relevance values. As shown in Fig. 4.8, two points (black 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).

Example 2: synthetic Gaussian data.

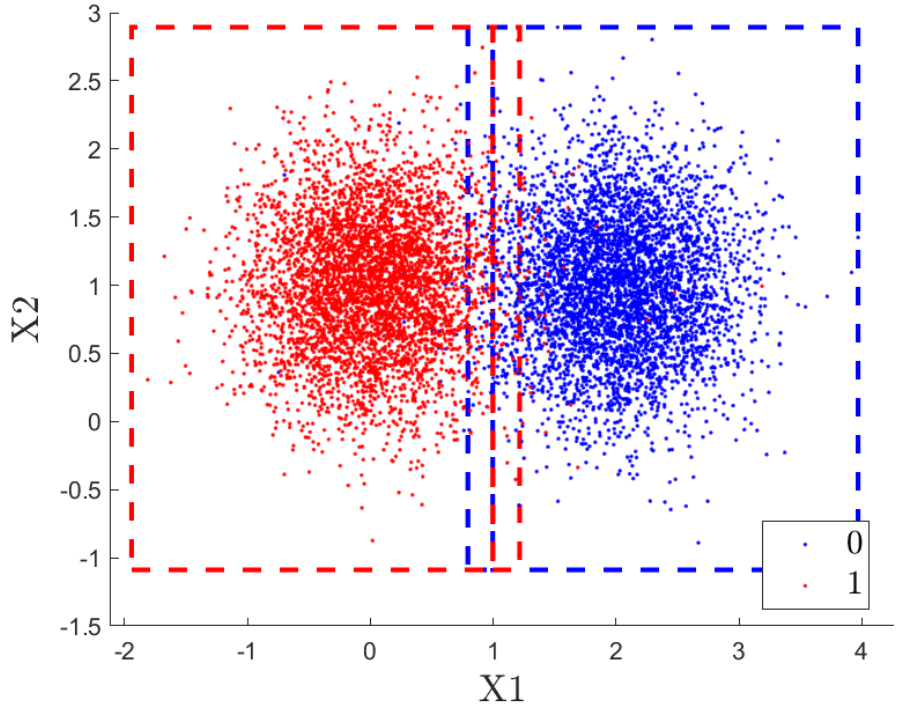


Figure 4.9: Scatter plot of 2D synthetic Gaussian data classified through rules (rectangles) via Logic Learning Machine model (see 2.2.3).

Another simplified example helping in making preliminary assessment of the CONFIDERAI approach involves the actual training of a rule-based model, the LLM (2.2.3), on bidimensional synthetic Gaussian data. These are generated to create a simplified scenario, where the ruleset being learned is expected to be short and pretty well-performing. In details, a binary classification dataset $\mathcal{Z}$ is built, by modelling the two classes as 2D Gaussian distributions:

$$\mathbf{X} \mid Y = y \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y), \quad \mathbf{X} = [X_1, X_2] \subset \mathbb{R}^2, \quad Y = \{0, 1\},$$

where

$$\boldsymbol{\mu}_0 = [2, 1]^T, \quad \boldsymbol{\Sigma}_0 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.8 \end{bmatrix}$$

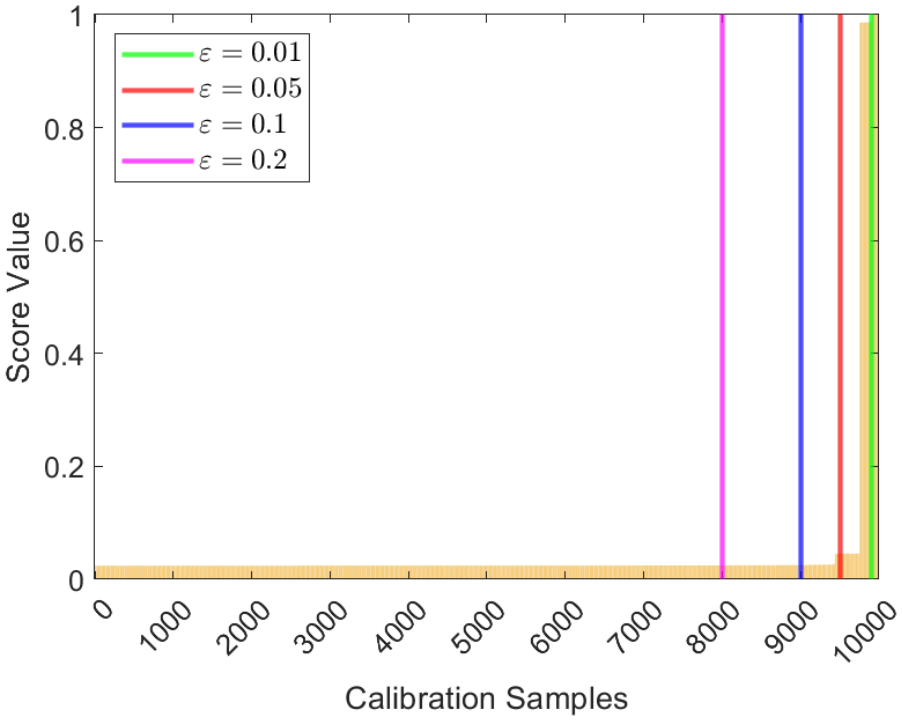


Figure 4.10: Ordered calibration scores and empirical quantiles at the following error levels: $\varepsilon = \{0.01, 0.05, 0.1, 0.2\}$.

and

$$\boldsymbol{\mu}_1 = [0, 1]^T, \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.8 \end{bmatrix}$$

By following the conformal prediction pipeline, dataset $\mathcal{Z}$ is split into three portions: i) a proper training set $\mathcal{Z}_{tr}$ of size $n_{tr} = 9000$, ii) a calibration set $\mathcal{Z}_c$ of size $n_c = 10000$, and a iii) test set of $\mathcal{Z}_{ts}$ of size $n_{ts} = 1000$. Rule generation via LLM model using $\mathcal{Z}_{tr}$ data provides the following ruleset $\mathcal{R}_{\text{gauss}}$:

r_1: **if** $X_1 > 0.9964$ **then** $y = 0$, $C(r_1) = 0.977$, $E(r_1) = 0.023$

r_2: **if** $0.7937 < X_1 < 0.9962$ **then** $y = 0$, $C(r_2) = 0.014$, $E(r_2) = 0.032$

r_3: **if** $X_1 \leq 0.9964$ **then** $y = 1$, $C(r_3) = 0.976$, $E(r_3) = 0.022$

r_4: **if** $0.9976 < X_1 < 1.2151$ **then** $y = 1$, $C(r_4) = 0.015$, $E(r_4) = 0.034$

As it can be seen from the covering $C(\cdot)$ and error $E(\cdot)$ metrics, each class is described by one main rule with high covering and low error (r_1 for class $y = 0$ and r_3 for $y = 1$), and an additional small covering rule (r_2 for class $y = 0$ and r_4 for $y = 1$). The simplicity of these rules, which only explicit the X_1 variable, reflects the separability of the class distributions, that only vary by a translation factor.

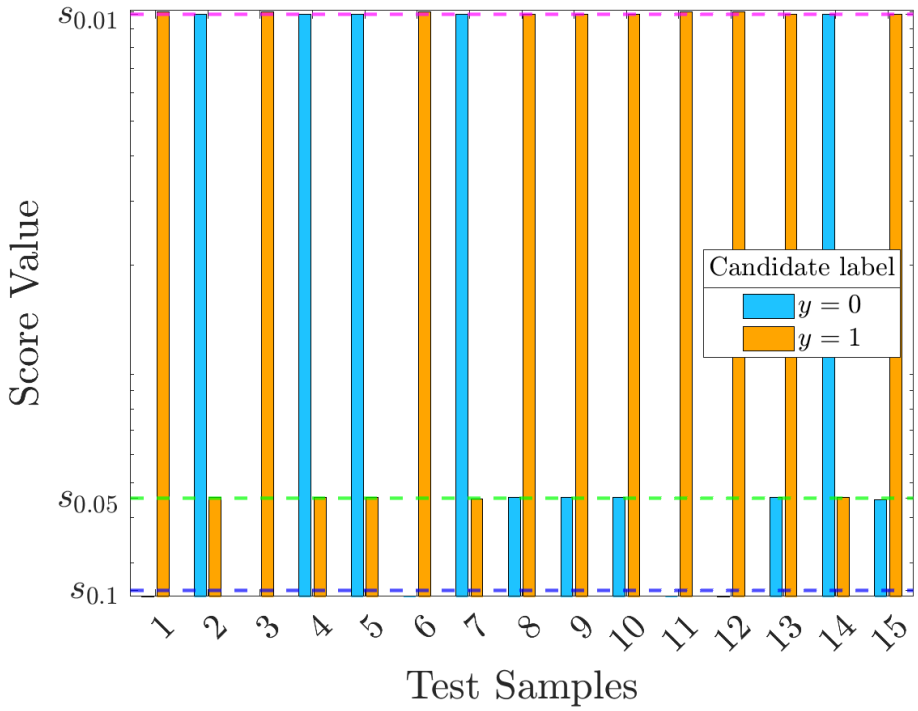
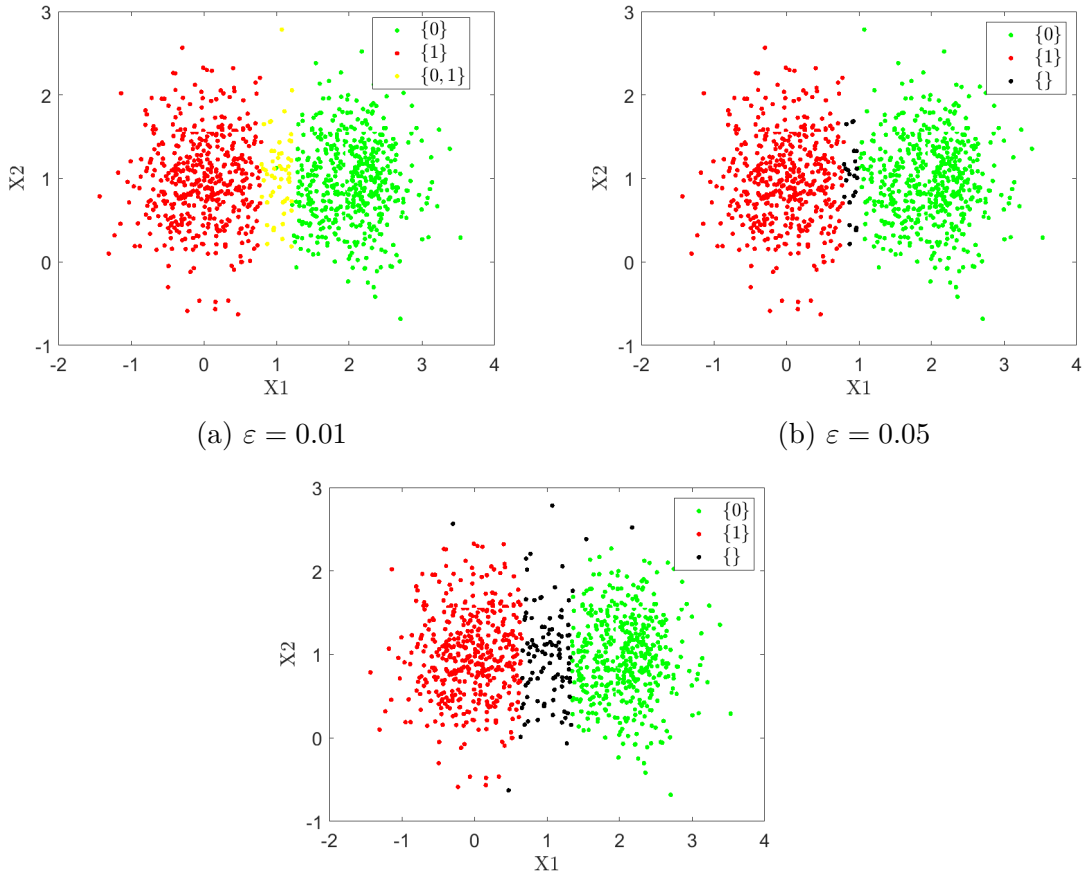


Figure 4.11: Test set scores compared to the calibration set scores empirical quantiles s_ε at different error levels ($\varepsilon = \{0.01, 0.05, 0.1\}$), for each candidate label ($y = 0$, light blue bars, or $y = 1$, orange bars).

This is also visible in Fig. 4.9, showing the scatter plots of the two classes along with dashed rectangles representing the boundaries of the learned rules (red for those predicting $y = 1$, blue for $y = 0$).
The score values for calibration $\mathcal{Z}_c$ and test $\mathcal{Z}_{ts}$ data are computed by applying Eq. 4.23. Prediction sets $\mathcal{C}_\varepsilon(\boldsymbol{x})$ for each $\boldsymbol{x} \in \mathcal{Z}_{ts}$ are then retrieved following Eq. 4.8. The empirical quantile s_ε at any desired ε level is computed based on ordered calibration score values (see Fig. 4.10 for $\varepsilon = \{0.01, 0.05, 0.1, 0.2\}$), and used as a decision criterion, acting like a threshold on test set score values, to define which labels should be included in $\mathcal{C}_\varepsilon(\boldsymbol{x})$.
Figure 4.11 illustrates this process for a subset of 15 test samples, by showing the scores for both candidate labels $y = 0$ and $y = 1$: the scores computed over test set samples by assigning the candidate label are thresholded via the s_ε values at three different levels of $\varepsilon = \{0.01, 0.05, 0.1\}$. This example points out the effect of choosing different ε on the prediction set. Considering, e.g., test sample 2, it will be: $\mathcal{C}_{0.01}(\boldsymbol{x}) = \{0,1\}$, $\mathcal{C}_{0.05}(\boldsymbol{x}) = \{1\}$, and $\mathcal{C}_{0.1}(\boldsymbol{x}) = \{\}$. The lower is ε the larger is quantile level (which is approximately $1 - \varepsilon$), thus increasing the chance of including that label in the prediction set. Conversely, higher values of ε tend to

exclude the label from the set.



(a) $\varepsilon = 0.01$

(b) $\varepsilon = 0.05$

(c) $\varepsilon = 0.1$

Figure 4.12: Scatter plots of the prediction regions obtained after applying CONFIDERAI score function to synthetic Gaussian data, for different error levels ($\varepsilon = \{0.01, 0.05, 0.1\}$).

Such behavior results in prediction sets containing both labels for the lowest ε, for points located in the regions where the two classes intersect (Fig. 4.12a); as soon as the error level is allowed to be higher, i.e. with growing ε, some points that were previously associated to double-sized prediction sets become singleton and, more often, empty sets (Figs. 4.12b-4.12c).

4.2.3 Comparisons with other methods

For the evaluation and a better understanding of the innovative aspects of CONFIDERAI score function, comparisons to other two kinds of score functions are considered: i) the *inverse probability*, which is the most canonical choice for a score

function in classification [12]; ii) a $k-$Nearest Neighbors score function, inspired, and partially adapted to the settings of CONFIDERAI, to the approach proposed in [3].

Inverse Probability Score

This score function is widely used in CP for classification problems, and it directly exploits the predicted class probabilities computed by a classifier to define a measure of non-conformity for the input samples. Specifically, given an instance $\boldsymbol{x}$ and a label y, it is computed as:

$$s(\boldsymbol{x}, y) = 1 - \Pr\{y|\boldsymbol{x}\}, \tag{4.24}$$

being $\Pr\{y|\boldsymbol{x}\}$ the posterior class probability of label y. Its computation depends on the specific rule-based classifier.

K-Nearest Neighbors Score

While the inverse probability score is used for any classifier (rule-based or not), CONFIDERAI is also compared to the work by Abdelqader et al. [3], which is tailored for rule-based classifiers. To make a fairer comparison with CONFIDERAI, some adaptations are needed:

1. A rule ordering needs to be imposed: for this, rule relevance metric (Equation 2.4) is considered. Therefore, each point is assumed to satisfy just one rule corresponding to the largest rule relevance.

2. Paper [3] applies the CP method using the definitions by Shafer & Vovk [130] involving p-values and non-conformity scores; since CONFIDERAI is based on the score function and quantile as per Angelopoulos et al. [11], the approach in [3] was still considered, but using the same setting as for CONFIDERAI.

Based on this, the proposed adaptation of [3] algorithm involves the steps summarized in the following, defining the approach referred to as the KNN score. Let $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$ be a proper training set, $\mathcal{X}_{cal} \times \mathcal{Y}_{cal}$ the calibration set, and $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$ the test set. Also, let $K \in \mathbb{N}$ be the number of nearest neighbors, and ε the error level. The KNN score approach consists in:

1. Train a rule-based classifier on $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$ and get a ruleset $\mathcal{R}$.

2. $\forall r \in \mathcal{R}$ do:

    a) Get subset of calibration points $\mathcal{X}^r_{cal} \subseteq \mathcal{X}_{cal}$ satisfied by r.

b) $\forall y \in \{0,1\}$, compute:

$$s(\boldsymbol{x}_i, y) = \frac{\sum_{j=1}^{K} d(\boldsymbol{x}_i, \boldsymbol{x}_j^y)}{\sum_{j=1}^{K} d(\boldsymbol{x}_i, \boldsymbol{x}_j^{\neg y})} \quad \forall \boldsymbol{x}_i \in \mathcal{X}_{cal}^r, \tag{4.25}$$

with $d(\cdot,\cdot)$ being the Euclidean distance, $\boldsymbol{x}_j^y \in \mathcal{X}_{cal}$ belonging to class y, and $\boldsymbol{x}_j^{\neg y} \in \mathcal{X}_{cal}$ a point belonging to a class other than y.

c) Compute s_ε^r, i.e., the quantile at level $1-\varepsilon$ of the calibration scores computed at step 2b).

3. Get the prediction sets for test data, that is, $\forall r \in \mathcal{R}$ do:

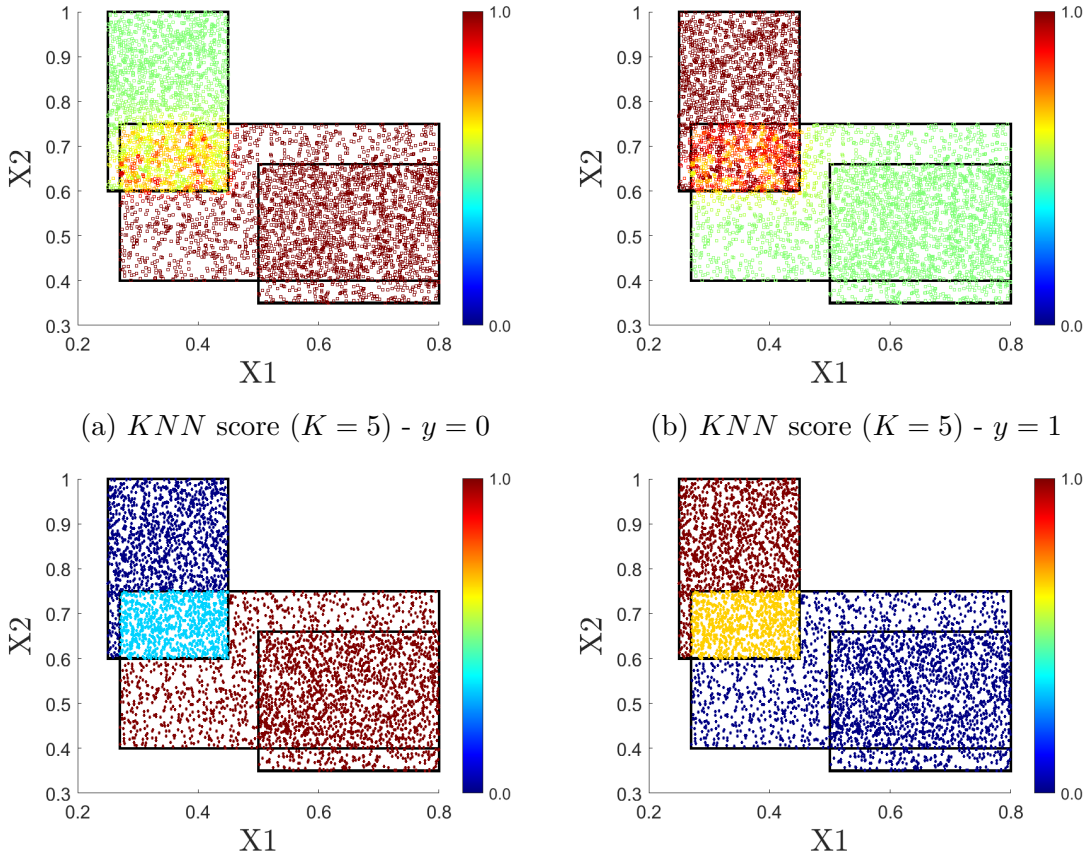
a) Get subset of test points $\mathcal{X}_{ts}^r \subseteq \mathcal{X}_{ts}$ satisfied by r.

b) Find the prediction set for points covered by rule r:

$$\mathcal{C}_\varepsilon^r(\boldsymbol{x}) = \{y \mid s(\boldsymbol{x}, y) \leq s_\varepsilon^r\} \quad \forall \boldsymbol{x} \in \mathcal{X}_{ts}^r$$

Comparison on toy rules

To compare the characteristics of CONFIDERAI versus the other two score functions, let us consider again the toy examples described in Section 4.2.2. In particular, to point out the key features of the three, the focus is posed only on the high overlap toy ruleset (r_1^{high}, r_2^{high}, r_3^{high}).
The plots of Figure 4.13 show the obtained values of the inverse probability (Figures 4.13c for label $y = 0$ and 4.13d for label $y = 1$) and the KNN score with $K = 5$ (4.13a and 4.13b for labels $y = 0$ and $y = 1$, respectively) on calibration data. By comparing these graphs with the corresponding ones in Figures 4.6e-4.6f, it can be seen that CONFIDERAI score is the only one that presents a variation of score values within the rule boundaries.
Another difference among the scores regards the way in which rule overlaps are considered. According to the inverse probability, in the overlap area at the intersection of two rules predicting opposite labels, i.e. r_1^{high} and r_2^{high}, the score is correctly higher than the score in the remaining area of r_1^{high} (in which no overlaps exist). In this same overlap region, CONFIDERAI is more conservative, by assigning even larger values. KNN score behaves in a similar way, but the values assigned to the overlap region (and, in general, to the whole r_1^{high}) are larger than those computed by inverse probability. Also, the KNN score function shows a differentiation of values within the overlap area that does not appear with the proposed score and inverse probability.
What is also important to observe is the different behavior when rules predicting the same label overlap: while inverse probability and KNN scores do not vary in the overlap region between r_2^{high} and r_3^{high}, CONFIDERAI score decreases, thus really accounting for both rules and making the prediction's confidence increase.



(a) KNN score ($K = 5$) - $y = 0$

(b) KNN score ($K = 5$) - $y = 1$

(c) inverse probability - $y = 0$

(d) inverse probability - $y = 1$

Figure 4.13: Scatter plots of the score values obtained by adopting the KNN score function with $K = 5$ nearest neighbors (top row) and the inverse probability (bottom row).

4.2.4 Conformal Critical Sets

Keeping in mind all the theory on how to set a conformal prediction (see 4.2.1), and relying on [26], the *conformal critical set* (CCS) at *confidence level* $1 - \varepsilon$ is defined as the subset $\mathcal{S}_\varepsilon \subseteq \mathcal{X}$, as follows:

$$\mathcal{S}_\varepsilon = \left\{ \boldsymbol{x} \mid s(\boldsymbol{x}, +1) \leq s_\varepsilon,\ s(\boldsymbol{x}, 0) > s_\varepsilon \right\}. \tag{4.26}$$

Put in words, the CCS is a subset of the input space where the prediction set is composed by only points belonging to the target class whose error is bounded by the CP, i.e. points $(\boldsymbol{x}, +1)$. This means that the model $\hat{f}$ is likely to make safe predictions for inputs in $\mathcal{S}_\varepsilon$ with a specified level of error ε.
In our context, we assume that the label $+1$ denotes the target class, which is to be

interpreted as the presence of *critical situations* in the system. The label 0 refers to the non-target class instead, which denotes the absence of such conditions.

Remark 4.2.2 (On the meaning of critical points)**.** *Note that the meaning of the term "critical" is context-dependent. For example, in a situation in which safety is of paramount importance, i.e. for instance in the case of collision avoidance, one may be interested in finding regions of the feature space where collision is avoided with high probability. In this case, the terms "critical" and "safe" may be seen as synonyms*[1]*. On the other hand, there are cases in which one would like to report only critical cases when the probability of failure is very high, so to avoid false alarms. In this case, the class* $+1$ *would correspond to the "failure" case.*

Following Equation 4.26, test points belonging to the CCS can be labelled as *conformal-critical*, providing a new way to look at the dataset and obtain a (good) approximation of the CCS, still preserving the interpretability. The process for this phase is given in Algorithm 2. Let us indicate with n_+ the number of conformal-critical points (i.e. labelled with $+1$) in the calibration set, and with n_0 the non-critical ones. Then, it is: $n_c = n_+ + n_0$.

Algorithm 2 Rule Correction based on CCS

Input: Critical class calibration set $(\boldsymbol{x}_1, +1), (\boldsymbol{x}_2, +1), \ldots, (\boldsymbol{x}_{n_+}, +1)$, conformal critical set $\mathcal{S}_\varepsilon$.
Output: $\mathcal{R}_{\mathcal{S}_\varepsilon}$.

1: Classify

$$\boldsymbol{x}_i \longrightarrow \begin{cases} +1 & \text{if} \quad \boldsymbol{x}_i \in \mathcal{S}_\varepsilon \\ -1 & \text{otherwise} \end{cases}$$

and create a new dataset

$$\mathcal{X} \times \tilde{\mathcal{Y}} = \{(\boldsymbol{x}_i, \tilde{y}_i) \mid \tilde{y}_i \in \{-1, +1\}\}\,.$$

2: Train a rule-based classifier on $\mathcal{X} \times \tilde{\mathcal{Y}}$ (e.g. LLM).
3: Get the new rules for $\tilde{y} = +1$ and output $\mathcal{R}_{\mathcal{S}_\varepsilon}$.

This procedure allows identifying new rules that characterize the CCS boundaries, which 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.

[1] In this case, the label $+1$ will denote no collision, and 0 will correspond to collision.

Practical implications

Some examples may help understand the practical impact of deriving critical sets. In case of a dynamical system (e.g., robots moving in a given environment, vehicle approaching specific maneuvers), the critical region may represent the subsets of states in which the application is considered safe (e.g., collision avoidance states, as to the vehicle platooning considered in the performance evaluation section). Finding those subsets may be a hard task in many practical situations and only data driven solutions are applicable (e.g., platooning is again a good example in that respect). On the other hand, those solutions need some guarantee and this is where the conformal framework comes into play. It consists of finding the regions with zero (or at least controllable) false negatives of (predicted) safety states, namely, the zones in which the dynamical system may move without experiencing any danger (such as collision with other agents or obstacles). The conformal critical set gives such an assurance. The safety engineer knows that the AI application has been properly designed as soon as the trajectories lye within that set. Characterizing the boundaries of the set is even more important in order to trigger appropriate alarms before any danger may take place.
Other examples may be provided with respect to the control of false positive rate. A first example deals with cybersecurity. In many cases, the alarms inherent to ongoing attacks lead to severe digital (sometimes even physical) countermeasures, such that security analysts want to be sure that an attack is really in play before unleashing all the necessary restrictions. In that respect, the conformal critical set profiles the conditions of the system (a network or a critical infrastructure) surely associated with the presence of attack (i.e., zero false positive). A second example deals with disease diagnosis by AI, as being made on the basis of clinical data over a population of patients (in comparison with healthy individuals). In this case, positive answer by the AI means the disease is predicted; false positive means the prediction was not realistic (e.g., after additional exams, the inauspicious diagnosis becomes invalidated). Circumventing the cases in which patients are surely affected by the disease is of great interest, also in respect to differentiating them with respect to the other cases where the disease prediction does not lie in the conformal borders. In the latter situation, additional exams should be even more urgent to settle the matter (disease or not disease).

Illustrative Example with Gaussian data

Based on the same synthetic Gaussian data as in Example 2 of Section 4.2.2, this example shows the application of Algorithm 2 in a simplified context. Class $y = 0$ points from Figure 4.9 are here assumed as the critical class (labelled $+1$ in the algorithm). Considering $\varepsilon = 0.01$, CONFIDERAI algorithm is applied to ruleset $\mathcal{R}_{\text{gauss}}$, and Equation 4.26 is used to determine the related conformal critical set. Then, according to Equation 2, points are re-labeled based on whether they belong

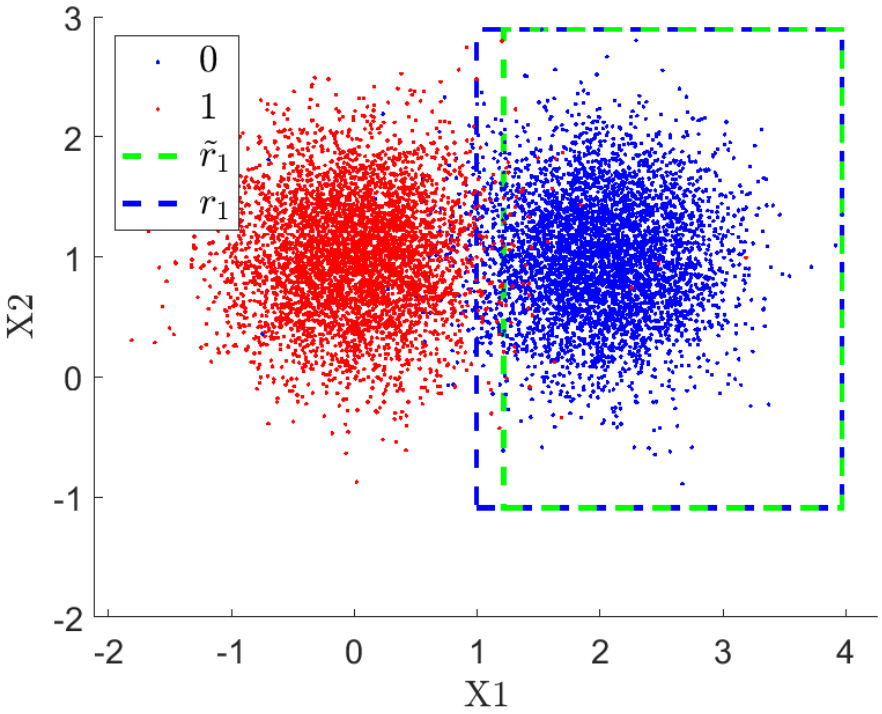


Figure 4.14: Example on 2D synthetic Gaussian data showing the comparison between the original most relevant rule r_1 (blue rectangle) and the optimized one $\tilde{r}_1$ after data re-labeling from conformal critical set.

to such set or not. Finally, the Logic Learning Machine (2.2.3) model is trained on the newly labeled dataset, generating the following rules for the critical class:

$\tilde{r}_1$: **if** $X_1 > 1.2154$ **then** $\tilde{y} = 1$, $C(\tilde{r}_1) = 0.92$, $E(\tilde{r}_1) = 0$

$\tilde{r}_2$: **if** $1.65 < X_2 < 1.97$ **then** $\tilde{y} = 1$, $C(\tilde{r}_2) = 0.07$, $E(\tilde{r}_2) = 0.05$

where covering and error values are computed on a test set. Compared to the original rules r_1 and r_2 from $\mathcal{R}_{\text{gauss}}$, the error of the first rule becomes 0 by introducing a modification of the threshold value for feature X_1 from 0.9964 to 1.2154, that slightly decreases the covering (from 0.97 to 0.92). Figure 4.14 well depicts this behavior, showing how $\tilde{r}_1$ encloses less points from the non-target class $y = 1$ by moving X_1 lower rule boundary to the right.
In contrast, the second rule $\tilde{r}_2$ introduces a restriction to variable X_2 values that was not present in the original ruleset, but it is however denoted by a very low covering and it can be disregarded. Overall, the new ruleset (cfr. original ruleset) reaches a TPR value of 0.91 (cfr. 0.95), precision of 1 (cfr. 0.96), and a F1 score of 0.95 (cfr. 0.96).

4.2.5 Experiments with real datasets

This Section presents the results of the experiments devoted to test the proposed score function, both in terms of canonical metrics in conformal prediction evaluation (i.e., accuracy, efficiency and computational time, see Sec. 4.2.5) and of conformal critical set (Sec. 4.2.5). The rule-based model adopted here is the Logic Learning Machine, LLM 2.2.3.
All the experiments were executed on a host equipped with Intel Core i5 dual-core processor at 2.6GHz and 8GB RAM memory. The host runs macOS version 11.7.10.

Datasets description

The experiments involved 10 different datasets from several fields, representing important scenarios from safety perspectives, open-source available and shortly described in the following:

- **P2P** and **SSH**: two datasets concerning peer-to-peer (P2P) and secure shell (SSH) applications of a Domain Name Server (DNS) tunneling detection system [5]. In both cases, data involve three physical quantities characterizing the communication: the size of query packets q, the size of answer packets a, and the time interval intercurring between query and answer Dt. For each of these measurements, statistical features (mean m, variance v, kurtosis κ and skewness s) were extracted over the time series of the system, thus leading to 12 features. The output variable is a binary label indicating if the sample is 'legitimate' or 'attack'.

- **BSS**: the Body Signals of Smoking dataset [19] collects personal and biological measurements from a group of subjects, with the aim of predicting if these quantities can represent biomarkers of *smoking* or *non-smoking* habits. Specifically, 25 features are collected: gender, age, height, weight, waist circumference length, left and right eyesight, left and right hearing, systolic and diastolic blood pressure, Fasting Blood Sugar, total, High-Density Lipoproteins (HDL) and Low-Density Lipoproteins (LDL) cholesterol, trygliceride, hemoglobin, urine proteins concentration, serum creatinine level, Aspartate Transaminase (AST), Alanine Transaminase (ALT), Guanosine Triphosphate (GTP), oral status (expressed as binary, good or bad), presence of caries, presence of tartar. Two classes, smoking or non-smoking, are present in a quite balanced proportion (63% non-smoking, 47% smoking).

- **CHD**: the Cardiovascular Heart Disease dataset [22] collects a set of 13 personal (age, gender, height, weight), clinical (systolic and diastolic blood pressure, total cholesterol, blood glucose) and behavioural (smoking/non-smoking status, alcohol consumption or not, physical activity/inactivity) features from

a group of patients. The aim is to understand whether a subject is affected by a cardiovascular disease or not.

- **Vehicle Platooning**: this dataset is the same already introduced in 3.3.1 and 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**: This dataset is hosted in NASA repository [139] and includes four different datasets corresponding to four engines of the same manufacturers. Each of them contains 23 physical quantities over time, and the first four statistical moments are extracted from them (similarly to the DNS case). Since working with 23×4 features was too complex, a preliminary analysis through LLM feature ranking (2.9) was performed to select a smaller set of 7 most important features, namely: s_{os2}, m_{Nc}, v_{Nc}, v_{phi}, $m_{htBleed}$, $s_{htBleed}$, m_{W31}. These refer to the following measurements on the system: $os2$ is the operational setting 2; Nc the physical core speed; phi the ratio of fuel flow to $Ps30$; $htBleed$ the bleed enthalpy; $W31$ the HTP coolant bleed. The output variable is linked to the RUL value, which represents the time before the occurrence of a fault and is binarized to assume either value 'healthy'(RUL>150) or 'fault' (RUL≤150). A rule-based classifier such as the LLM is then used to predict if the engine would enter a faulty state or not.

- **EEG**: the Eye State Classification EEG dataset [41] reports the state of patients' eyes (open or closed) based on continuous electroencephalogram (EEG) measurements at different electrode positions acquired via a Emotiv EEG Neuroheadset device.

- **MQTTset** [140]: based on Message Queue Telemetry Transportation communication protocol, this dataset collects measurements about MQTT communication quality from different Internet of Things devices to simulate a smart home environment; cyber-attacked data are also included to detect *malicious* and *legitimate* traffic.

- **Magic**: the Magic Gamma Telescope dataset [86] reports Monte Carlo simulations of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope to distinguish between gamma and hadron radiation.

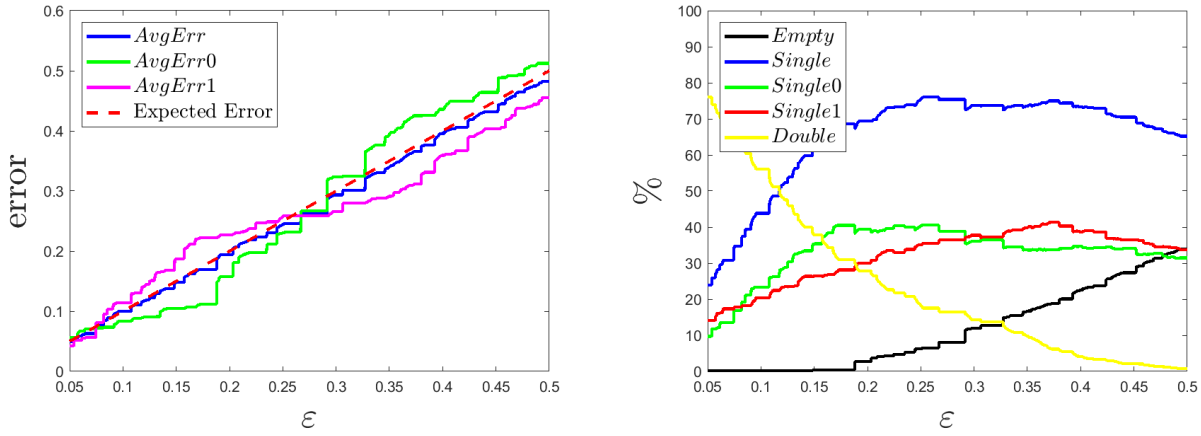
- **Fire Alarm**: this dataset [131] contains data to develop an AI-based smoke detection device, involving a binary target, related to *presence* or *absence* of fire.

Table 4.2: Evaluation metrics (error and size, Sec. 4.2.1) for CONFIDERAI on Logic Learning Machine model tested on the 10 chosen datasets.

	Time_{nC}[s]		Error			Size		
			avgErr	*avgErr0*	*avgErr1*	*avgEmpty*	*avgSingle*	*avgDouble*
P2P	160	$\varepsilon = 0.01$	0	0.001	0	0	0.946	0.054
		$\varepsilon = 0.05$	0.039	0.079	0	0.026	0.966	0.008
		$\varepsilon = 0.1$	0.048	0.079	0.018	0.035	0.965	0
		$\varepsilon = 0.2$	0.152	0.226	0.077	0.152	0.848	0
SSH	200	$\varepsilon = 0.01$	0.008	0.004	0.011	0	0.645	0.354
		$\varepsilon = 0.05$	0.039	0.015	0.063	0.005	0.761	0.234
		$\varepsilon = 0.1$	0.094	0.05	0.139	0.039	0.822	0.139
		$\varepsilon = 0.2$	0.193	0.106	0.28	0.112	0.83	0.058
BSS	340	$\varepsilon = 0.01$	0.01	0.005	0.019	0	0.238	0.762
		$\varepsilon = 0.05$	0.051	0.068	0.021	0.001	0.354	0.645
		$\varepsilon = 0.1$	0.101	0.127	0.053	0.008	0.484	0.508
		$\varepsilon = 0.2$	0.184	0.203	0.149	0.043	0.641	0.315
CHD	150	$\varepsilon = 0.01$	0.012	0.018	0.006	0.001	0.072	0.927
		$\varepsilon = 0.05$	0.049	0.057	0.042	0.001	0.239	0.76
		$\varepsilon = 0.1$	0.1	0.084	0.114	0.001	0.438	0.56
		$\varepsilon = 0.2$	0.194	0.158	0.227	0.027	0.694	0.279
Vehicle Platooning	133	$\varepsilon = 0.01$	0.012	0.02	0.003	0	0.536	0.464
		$\varepsilon = 0.05$	0.052	0.044	0.063	0.004	0.76	0.237
		$\varepsilon = 0.1$	0.102	0.099	0.104	0.034	0.844	0.122
		$\varepsilon = 0.2$	0.208	0.185	0.238	0.136	0.835	0.03
RUL	127	$\varepsilon = 0.01$	0.025	0.028	0.016	0.005	0.398	0.598
		$\varepsilon = 0.05$	0.058	0.06	0.051	0.007	0.557	0.436
		$\varepsilon = 0.1$	0.101	0.096	0.111	0.013	0.707	0.28
		$\varepsilon = 0.2$	0.191	0.185	0.206	0.066	0.807	0.126
EEG	300	$\varepsilon = 0.01$	0.02	0.015	0.026	0.001	0.327	0.672
		$\varepsilon = 0.05$	0.065	0.053	0.08	0.008	0.487	0.505
		$\varepsilon = 0.1$	0.097	0.093	0.102	0.021	0.565	0.414
		$\varepsilon = 0.2$	0.19	0.194	0.185	0.083	0.67	0.247
MQTTset	220	$\varepsilon = 0.01$	0.009	0.004	0.015	0	0.705	0.295
		$\varepsilon = 0.05$	0.04	0.009	0.07	0.006	0.826	0.168
		$\varepsilon = 0.1$	0.053	0.009	0.097	0.006	0.925	0.069
		$\varepsilon = 0.2$	0.167	0.091	0.24	0.144	0.794	0.062
Magic	200	$\varepsilon = 0.01$	0.025	0.012	0.048	0.001	0.329	0.67
		$\varepsilon = 0.05$	0.066	0.037	0.116	0.004	0.615	0.381
		$\varepsilon = 0.1$	0.13	0.116	0.155	0.039	0.688	0.273
		$\varepsilon = 0.2$	0.222	0.206	0.249	0.111	0.774	0.115
Fire Alarm	132	$\varepsilon = 0.01$	0	0	0	0	0.973	0.027
		$\varepsilon = 0.05$	0.011	0	0.022	0	0.985	0.015
		$\varepsilon = 0.1$	0.077	0.133	0.022	0.077	0.91	0.013
		$\varepsilon = 0.2$	0.166	0.314	0.022	0.166	0.834	0

Score function evaluation

For the evaluation of CONFIDERAI score function, the average errors and efficiency metrics defined in Equations 4.10-4.14 were computed, at different choices of the ε level, namely $\varepsilon = 0.01$, $\varepsilon = 0.05$, $\varepsilon = 0.1$ and $\varepsilon = 0.2$. To provide an indication on the time complexity of the method, the time spent for computing the scores on the calibration set, whose size n_C was set to 10000 for all datasets, was also calculated. The obtained results are reported in Table 4.2.
The overall metrics on the benchmark datasets outline the expected behavior of the conformal prediction. Average error trend follows the values of ε in all cases, confirming that CONFIDERAI score yields *valid* prediction sets. In the P2P and Fire Alarm datasets it is possible to observe that the $avgErr$ is well lower the ε value, and the reasons explaining this behavior can be found in the relative simplicity of these classification problems. The original LLM models were indeed made up of a small set of rules (6 rules in both cases) and achieved a very high performance (accuracy higher than 98% for both).
If, on the one hand, CP validity is guaranteed from CP theory itself, CP efficiency provides a way to assess whether the prediction sets yielded by the introduced score are useful enough. As anticipated in Sec. 4.2.1, efficiency is measured through the size of prediction sets. To satisfy the marginal coverage guarantee (4.9), the most trivial solution would be indeed to include both available labels in the prediction set, thus contributing to the *avgDouble* metric, but this would obviously result in uninformative prediction sets. In contrast, if just one label was included in the prediction set, the performance guarantee would be reached in a more efficient way, since only one alternative would be provided and CP would ensure that a level $1 - \varepsilon$ would be maintained for this prediction. Regarding the size metrics ($avgEmpty$, $avgSingle$ and $avgDouble$) shown in Table 4.2, it is possible to notice that the empty prediction set rates grow with ε, and that the double-sized regions decrease with it. This is a coherent behavior, since at low ε the CP algorithm tends to put more labels in the prediction set to maintain the error under that low bound. Conversely, when the error level is allowed to increase, the algorithm reduces the two-label sets while the singleton predictions increase, and the empty ones increase. As an example, the metrics' trend for the CHD dataset is illustrated in Fig. 4.15, for $\varepsilon \in [0.05, 0.5]$. It can be observed that the average error on class 0 , i.e. the *healthy* samples, is lower than the average error on class 1, i.e. the *disease* class, up to an ε value around 0.27; for values larger than this value the trend reverses. Concerning the size, it can be notice that for about $\varepsilon = 0.12$ the singleton and double-size prediction regions occur in the same percentage (around 50%), with no empty predictions. For larger ε, singleton predictions rate continues to grow up to about $\varepsilon = 0.33$, where the empty regions increase (with double-sized sets continuing to decrease).
The computation time across the calibration sets is on average 196±73 s. Its



(a) Average errors

(b) Average sizes

Figure 4.15: Trend of the performance metrics obtained on the CHD dataset by varying $\varepsilon \in [0.05,\ 0.5]$

variation is mainly due to two factors: on the one hand, the number D of features of the dataset influences both the computational cost of the geometrical rule similarity ratio and the γ terms of Eq. 4.18, resulting in larger times for datasets with more dimensions, such as the BSS for which $D = 19$; on the other hand, the number of rules M_r generated by the LLM also influences the computation of the score: the larger is M_r, the higher is the chance to have multiple rules covering the same samples, resulting in more terms to be multiplied in Eq. 4.23. For example, the quickest computation (127 s) was achieved on the RUL dataset which has $D = 7$ and $M_r = 42$ rules, while the CHD took a longer time (150 s) having the same D but $M_r = 72$ (i.e., almost 80% more rules).

Comparison with other score functions

A further evaluation of CONFIDERAI score regarded the comparison with the inverse probability and KNN score functions described in Section 4.2.3. Since, in the CP framework, the average error (*avgErr*) is guaranteed to be bounded by ε by design, the comparison is carried out in terms of efficiency. Figures 4.16 and 4.17 report the *avgSingle* and *avgDouble* metrics for $\varepsilon = 0.05$ for all ten datasets. It is possible to observe that the performance of CONFIDERAI is similar to that of the inverse probability score, that is, generally good efficiency (being *avgSingle* > *avgDouble*). In contrast, the KNN score with $K = 5$ results in less efficient prediction sets, with lower *avgSingle* and larger *avgDouble* than the achieved values for the other two score functions. Despite the comparable results between CONFIDERAI and the canonical score function for classification, I would remark again the differences highlighted in the toy examples (4.2.3) in terms of how CONFIDERAI design

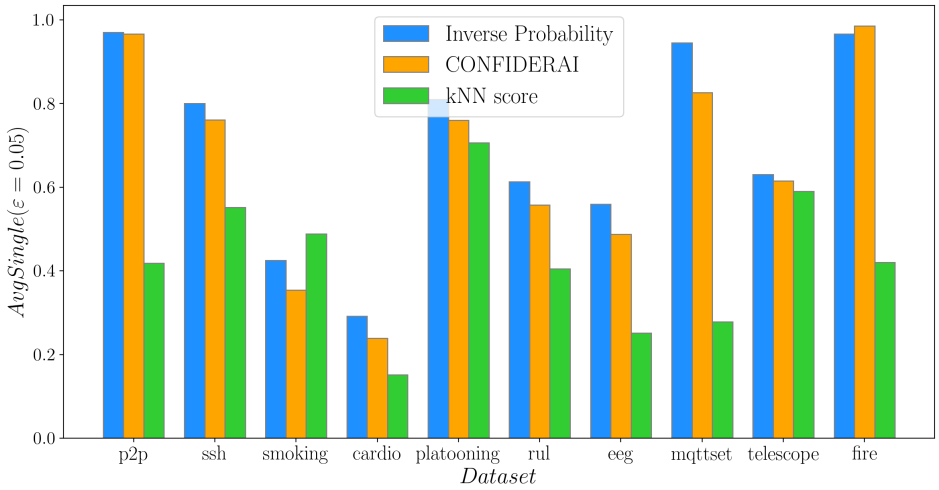


Figure 4.16: Comparisons of CONFIDERAI with inverse probability and KNN scores based on *avgSingle* metric, for all datasets at $\varepsilon = 0.05$.

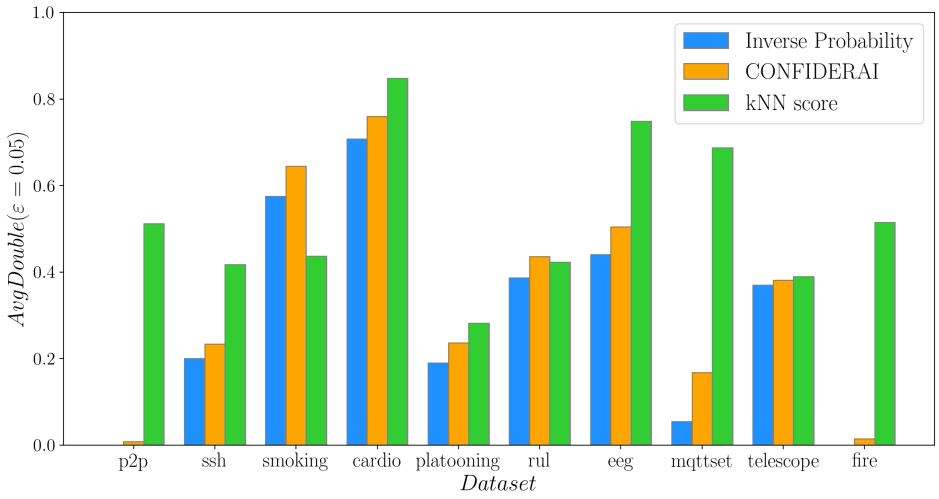


Figure 4.17: Comparisons of CONFIDERAI with inverse probability and KNN scores based on *avgDouble* metric, for all datasets at $\varepsilon = 0.05$.

choice allows to also consider the geometrical structure of each rule, while inverse probability only accounts for inference performance.

Evaluation of Conformal Critical Sets

As per Algorithm 2, the LLM model was trained again over the newly labelled datasets, thus finding new rulesets $\mathcal{R}_{\mathcal{S}_\varepsilon}$ for each dataset, whose rules predicting label $+1$ provide an interpretable description of the CCS. For this phase, it was decided to evaluate the case of $\varepsilon = 0.05$.

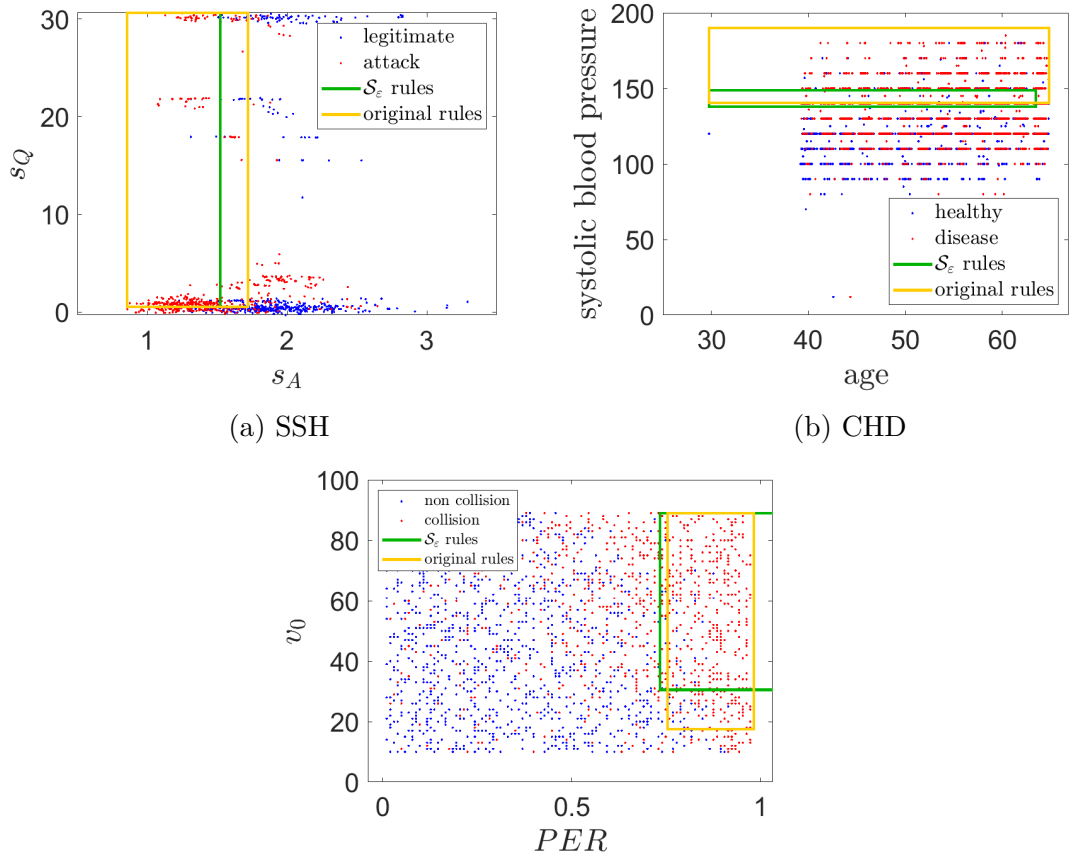
Table 4.3: Performance of the LLM trained with new labels from Eq.2

	TPR	**PPV**	**F1**
SSH	0.64	0.94	0.76
P2P	1.00	0.93	0.96
BSS	0.55	0.62	0.58
CHD	0.47	0.80	0.59
Vehicle Platooning	0.72	0.86	0.78
RUL	0.53	0.69	0.60
EEG	0.44	0.77	0.55
MQTTSet	0.89	0.92	0.90
Magic	0.63	0.90	0.73
Fire Alarm	0.88	0.98	0.93

The new rules were assessed with respect to the ground truth labels, in terms of true positive rate (TPR), precision (also known as PPV) and F1 score. The first two metrics were chosen as the only meaningful performance evaluations of $\mathcal{R}_{\mathcal{S}_\varepsilon}$ are referred to the critical class (i.e., $+1$ labels in this case), which is the target of the CCS. The F1 score measures their balance, being defined as their harmonic mean. Conversely, the metrics referring to the other class have been disregarded, since having $\tilde{y}_i = -1$ does not correspond to predictions for $y = 0$, but also for $y = 1$ points that do not achieve a singleton prediction region through the score function. Table 4.3 reports the metrics obtained for all the datasets. It can be observed that the TPR varies among the datasets, from lower values (e.g., EEG or CHD) to better ones (e.g., P2P or platooning), reflecting a different size (here intended as number of points enclosed) of the respective conformal critical sets. Indeed, TPR measures the ratio of correctly predicted $+1$ points with respect to all the points of that class. On the other hand, PPV values are overall high, exceeding the 80% in all cases except for BSS, RUL and EEG: this result suggests that the new rules well represent the critical class points, with just a few non-critical target points covered by these rules. Achieving both high TPR and PPV is a known challenge

in the precision-recall trade-off, however a good balance has been reached in the largest part of cases, as pointed out by the acceptable F1 score values (which are computed as the harmonic mean of TPR and PPV).

The metrics reported above refer to point predictions derived, in each dataset, from the different rules of the model, according to Eq. 2.6. Of course, the inspection of the single rules (predicting label +1) ends up with analogous conclusions.



(a) SSH

(b) CHD

(c) Vehicle Platooning

Figure 4.18: 2D scatter plots of three datasets, showing 2D boundaries of the most relevant rules from the original LLM classifier (yellow box) and from the new model derived via the conformal critical set ($\mathcal{S}_\varepsilon$ rule, green box).

Figure 4.18 shows 2D scatter plots of the classes of the SSH (Fig.4.18a), CHD (Fig.4.18b) and platooning (Fig.4.18c) datasets, where red points denote the critical classes ($y = +1$) to be characterized and blue points the non-critical ones ($y = 0$). The yellow boxes denote the first rule by relevance from the original model, and the green boxes the top-relevant new rules learned through $\mathcal{S}_\varepsilon$ (referred to as $\mathcal{S}_\varepsilon$ rule). In all three cases, it can be observed that the new rules leave some blue points

Table 4.4: Covering, error and precision of the most relevant rule predicting the critical class (+1) of the original LLM model and of the new $\mathcal{R}_{\mathcal{S}_\varepsilon}$ model, for the SSH, CHD and vehicle platooning datasets.

			Covering	**Precision**	**Error**
SSH	$\mathcal{S}_\varepsilon$ rule	**if** $s_A \leq 1.52 \wedge s_Q > 0.56$ **then** *attack*	0.45	0.95	0.03
	Original rule	**if** $v_A \leq 38058 \wedge v_Q \leq 2095 \wedge$ $s_A \leq 1.72 \wedge s_Q > 0.55$ **then** *attack*	0.35	0.91	0.03
CHD	$\mathcal{S}_\varepsilon$ rule	**if** *age* $\leq 63 \wedge$ *height* $> 152 \wedge$ *weight* $\leq 87 \wedge$ $139 <$ *systolic blood pressure* $\leq 149 \wedge$ *diastolic blood pressure* $> 79 \wedge$ *cholesterol* $\leq 2.5 \wedge$ *gluc* ≤ 2.5 **then** *disease*	0.13	0.89	0.02
	Original rule	**if** *systolic blood pressure* > 140 **then** *disease*	0.22	0.82	0.05
Vehicle platooning	$\mathcal{S}_\varepsilon$ rule	**if** $PER > 0.74 \wedge v0 > 30$ **then** *collision*	0.37	0.88	0.05
	Original rule	**if** $PER > 0.76 \wedge v0 > 17$ **then** *collision*	0.44	0.86	0.07

out, thus increasing in precision, even if sometimes this dramatically reduces the covering, such as in the CHD case. However, it is a known trade-off in machine learning between precision and recall: dealing with critical problems, it is often acceptable to have small coverings when reducing the error and increasing precision as much as possible. And this is the behavior exhibited by the examples shown in the figures, whose structure and performance details are reported in Table 4.4. In the SSH dataset, the new most relevant rule brings higher precision, while the error remains the same as in the original rule. In the CHD and vehicle platooning, the rules retrieved by the CCS result in both precision and error improvements with respect to the original case. These examples also reveal different possibilities that can occur when moving from the original ruleset to the new one. In some cases, the new rules may exclude the role of some features that previously had one, and this kind of guidance helps filtering out variables of the problem at hand that have no impact on the conformal guarantees: it is the case of the SSH, where features v_A and v_Q are no more present in the premise of $\mathcal{S}_\varepsilon$ rule. In some other cases, such as the platooning in the proposed example, the conditions remain exactly on the same variables, but with changes to the related thresholds. Finally, it can happen that $\mathcal{S}_\varepsilon$ rules also reveal new factors that were not highlighted in the original rules, as for in the CHD example, where it can be observed that the $\mathcal{S}_\varepsilon$ rule expresses much more conditions (on different features) than the original one: in a real clinical application, this might serve to clinicians as a more thorough guidance, revealing all the factors most probably involved in the presence of a cardiovascular disease. These are therefore short yet important examples motivating the introduction of conformal critical sets to shed light into how rule-based classifiers can be tuned via conformal prediction guarantees to achieve rules with higher guarantees on a target (critical) class.

4.2.6 Conclusions

This Section introduced CONFIDERAI, a new score function for rule-based binary classifiers directly designed on top of the properties of these models. Indeed, starting from the decision rules generated by the model, conformity is derived as a function of the placement of the samples with respect to the geometry of the model, also taking into account rule relevance, a measure that reflects the predictive quality of a rule. Moreover, the score takes into account possible rule overlaps thanks to the geometrical rule similarity metric.
Extensive experimentation by considering the Logic Learning Machine model on several datasets has shown a behavior in line with conformal prediction framework, both in terms of average errors and efficiency of the prediction sets. In addition, by leveraging on the results of CONFIDERAI, a step beyond the probabilistic guarantees provided by the conformal predictions has been reached, in the direction of a more actionable safety-preserving solution. That is, the notion of conformal critical set first guarantees to efficiently predict the critical class points (i.e., the target ones) in high probability (thanks to the CP); also, it allows retraining the rule-based classifier on a newly labeled version of the data, ending up with rules characterized by improved precision and reduced false positives on the critical class. By providing the tools for controlling the misclassfication error while keeping the explainability of the model, this result is a relevant achievement in the way to intersect explainable and reliable AI.

Chapter 5

Relevant applications in Industry 4.0

Combining explainability and reliability in ML tasks is of great interest in many fields, including cybersecurity, healthcare, and robotics. This Chapter presents some works where the techniques I introduced in my PhD career have been applied to wider problems: Section 5.1 shows the usage of the methods from Section 4.1 in the context of reliable detection of ***adversarial machine learning*** attacks; Section 5.2 describes how rule-based classification, along with syntactic rule similarity from Sec. 3.2 can reveal an innovative tool for the ***evaluation of synthetic data generation*** processes in the healthcare domain; Section 5.3 presents a work from REXASI-PRO research project I contributed to (A.1), showing the importance of the intersection between XAI and RAI in the context of ***robotics navigation***.

5.1 Explainable and Reliable Adversarial Machine Learning detection

5.1.1 Context and contribution

Adversarial machine learning (AML) is one of the main threats towards ML safety arisen in recent years, consisting in malicious attacks being performed against data. The main scope of these attacks is to inject malicious data into legitimate ones, with the aim of leading ML algorithms to failure, by generating misclassification or performance reductions [106]. The detection of these attacks is, to date, an important point to be addressed in fulfilling trustworthy AI principles [60], and it is challenging due to the minimal global perturbations characterizing adversarial attacks, that make their identification complex. AML is therefore a good scenario where reliable AI approaches from Chapter 5, combining explainability with error control, can find interesting applications [40]. This work thus investigates a

novel approach to the detection of AML attacks, going beyond traditional machine learning. In particular, the rule-based reliable AI techniques from Section 4.1 have been explored, by bounding the statistical classification error towards zero, while simultaneously maximizing the number of data points that exhibit this property. Emphasis is put on the explainability of the solution, which helps understand how the attacks work and identify the most sensitive areas in the feature space for the execution of such attacks.

5.1.2 Methodology

Attack phase

As the first step, Carlini-Wagner (CW) [28], Fast Gradient Sign Method (FGSM) [52], and Jacobian-based Saliency Map (JSMA) [106] attacks are executed against a victim ML model (see Fig. 5.1, attack phase), which in this case is a neural network of 3 layers with 512, 256 and 128 neurons, respectively. The network 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%.

Detection phase

The defensive strategy devised here operates as a robustness enhancement outside the main training model [40]. A binary classifier is trained on a dataset obtained by combining the original (*legitimate class*) and the adversarial data (*adversarial attack class*) and after labelling samples according to the presence or absence of adversarial attacks. The overall workflow is shown in the bottom part of figure 5.1. The detection classifier is designed to identify as many attacks as possible, while minimizing the False Positive Rate (FPR). In other words, the objective is to individuate data-driven *adversarial regions* where the presence of adversarial attacks is predicted with zero false positives. In this way, more attack data might be misclassified as legitimate (increase in false negative rate), but a good compromise is sought under the adopted Reliable AI methods. To this end, two groups of techniques have been adopted.

Rule-based reliability. On the one hand, methods that are inherently explainable being based on a rule-based classifier (the LLM 2.2.3), namely the *inside*, *outside* and *LLM0%* algorithms described in Sec. 4.1, which are now adapted to identify regions with controlled False Positives (FPs) while having the largest True Positives (TPs) as possible. This implies setting the class of the adversarial attack points as the target class in Algorithm 1. Apart from that, all the three methods follow the same procedures and equations as outlined in 4.1.

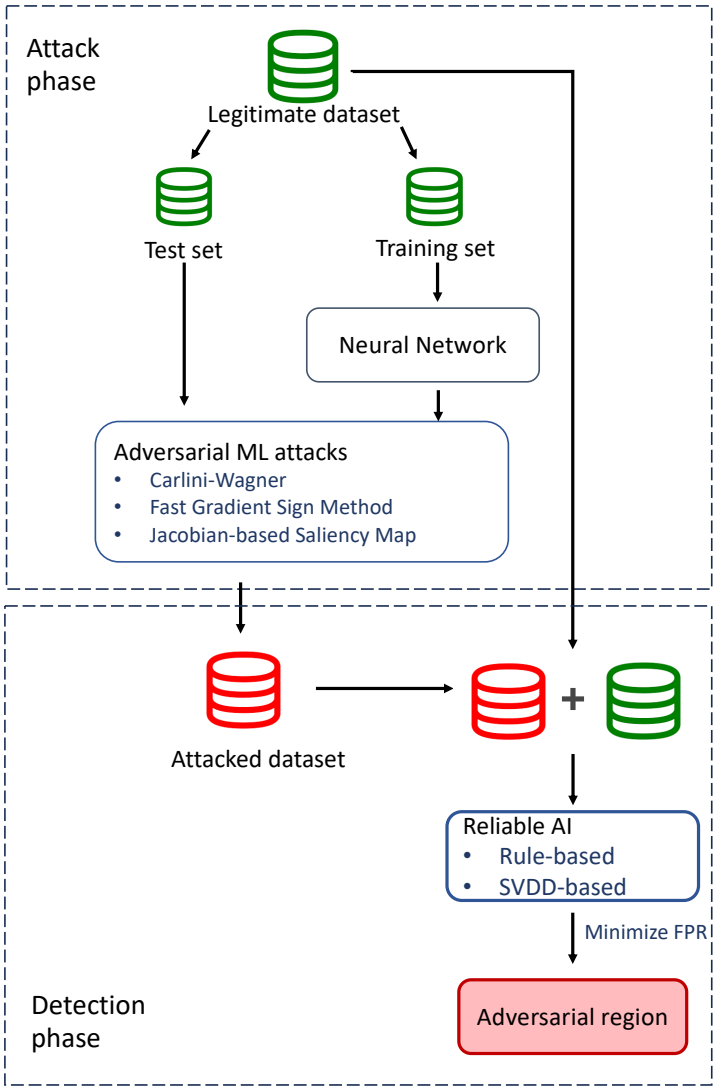


Figure 5.1: Scheme of the adversarial machine learning attack generation and detection phases. Attack phase: CW, FGSM and JSMA attacks target legitimate data, making a neural network failing. Detection phase: legitimate and attacked data are studied via Reliable AI classifiers for defining *adversarial regions* describing the attacked class with minimized error.

Safe Support Vector Data Description. In addition, another approach from [23] has been considered: this scheme, called *zeroFPRSVDD*, is based on an elaboration of the Support Vector Data Description (SVDD)[135, 136] framework. It

performs successive iterations of the SVDD on the initial target region, found with a preliminary SVDD, until there are no more legitimate points inside of it. The convergence is achieved when a fixed number of iterations is reached or when the desired value of FPR is satisfied. Moreover, rules are then extracted from the obtained adversarial region, by applying the LLM on a new dataset of instances sampled around the edge of the zeroFPRSVDD and classified through it, thus providing explainability to what was originally a black-box solution (*explainableSVDD*). More technical details on these methods are available in the related paper [23].

5.1.3 Experiments and main findings

Datasets

The proposed approach has been tested by considering three applications of interest: 1) DNS-SSH tunneling from network security field (see 4.2.5 for a description); 2) vehicle platooning (see 3.3.1); 3) RUL estimation, a benchmark in predictive maintenance (see 4.2.5 for a description). 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 the work will investigate how the different proposed attacks impact on this base performance. In this way, the detection methods will be tested on scenarios of increasing complexity.

Reliable AI performance

As mentioned, reliability from inside, outside and LLM0% (Sec. 4.1) algorithms have been applied for the detection of the adversarial attacks, with the objective of individuating adversarial regions with controlled FPR. Logic Learning Machine model (with 5% maximum error allowed for each rule) was then trained on the considered datasets, and feature/value ranking for $N_{FR} = 2$ allowed the selection of the original value ranking intervals for running inside and outside methods, as shown in Table 5.1.
The optimal solutions, i.e., the *adversarial regions* with minimized FPR with inside and outside methods are also reported in Table 5.1, where they can be compared with the original regions to understand how the thresholds change with the application of the algorithms. Their performance, in terms of confusion matrix metrics, is shown in Table 5.2, along with the other reliable AI techniques considered, i.e., the LLM0% and also the SVDD-based approaches adopted for comparison.
By looking at the results of rule-based techniques, it is possible to observe that the overall performance of both *inside* and *outside* methods is better on DNS tunneling

Table 5.1: **Inside, Outside**. Obtained adversarial regions with inside and outside methods in the three considered datasets, in comparison with the original intervals. These regions can provide insights about where, in the feature space, the adversarial attacks are more probably found.

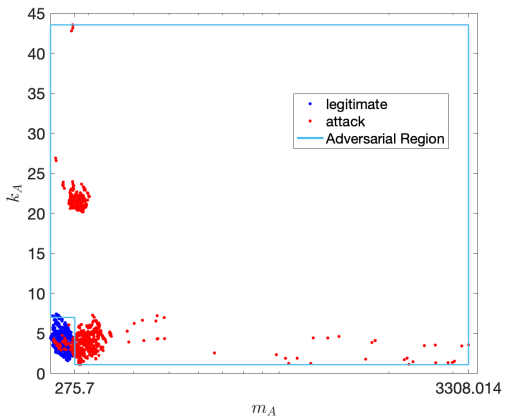
		DNS		**Platooning**		**RUL**	
		Original Intervals	**Adversarial Region**	**Original Intervals**	**Adversarial Region**	**Original Intervals**	**Adversarial Region**
Inside	CW	m_A>270.23 ∨ s_{Dt}>70.65	m_A>275.7 ∨ s_{Dt}>70.65	PER≤0.08 ∨ v_0>82	PER<0 ∨ v_0>89	$s\_htBleed$<0.39 ∨ s_os2<0.41	$s\_htBleed$<−0.817 ∨ s_os2<−0.558
	JSMA	$m_A > 270.10$ ∨ $k_A > 7.42$	m_A>275.7 ∨ k_A>6.99	d_0>8.97 ∨ $F_0 > -1$	d_0>8.99 ∨ $F_0 > -1$	v_phi>10.22 ∨ $s\_htBleed$>19.11	v_phi>0.262 ∨ $s\_htBleed$>0.875
	FGSM	s_A≤1.71 ∨ m_A>269.56	s_A≤1.63 ∨ m_A>270.9	N≥6 ∨ $F_0 < -8$	N≥10 ∨ $F_0 < -8$	v_phi>0.34 ∨ m_Nc<9048.38	v_phi>0.22 ∨ m_Nc<9038.35
Outside	CW	m_{Dt}>0.31 ∨ v_A>24503	m_{Dt}<0.34 ∧ v_A<25923	v_0≥82 ∨ d_0>8.67	v_0<43 ∧ d_0≤4.013	$m\_htBleed$<391.76 ∨ $s\_htBleed$<0.64	$m\_htBleed$>395.52 ∧ $s\_htBleed < -0.504$
	JSMA	$m_A \leq 270.34$ ∨ v_A>33393	$m_A > 276.58$ ∧ v_A<39286	$F_0 < -7$ ∨ d_0≤6	$F_0 \geq -4$ ∧ d_0≥8.99	v_phi<10.22 ∨ $s\_htBleed$<19.11	v_phi>0.144 ∧ $s\_htBleed$>1.172
	FGSM	s_A>1.82 ∨ m_A<267.92	s_A≤1.68 ∧ m_A>275.02	N≥9 ∨ PER<0.16	N≤3 ∧ PER>1	v_phi<0.34 ∨ $m\_htBleed < 395.76$	$v\_phi > 0.0796$ ∧ $m\_htBleed$>395.62

Table 5.2: **Inside, Outside, LLM0%**. Obtained performance metrics for the detection of adversarial attacks based on reliable AI.

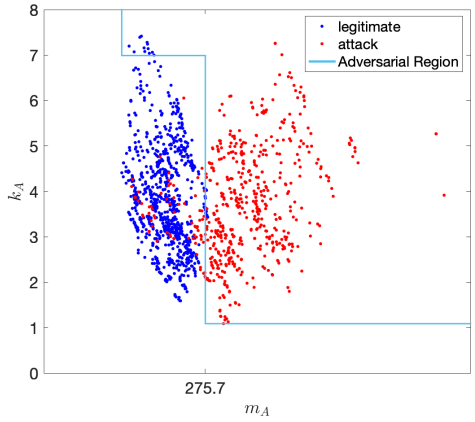
		DNS				**Platooning**				**RUL**			
		FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR	FPR	TPR	TNR	FNR
Inside	CW	0.03	0.45	0.97	0.55	0.03	0.02	0.97	0.98	0.02	0.01	0.98	0.99
	JSMA	0.03	0.93	0.97	0.07	0.07	0.56	0.93	0.44	0.02	1.00	0.98	0.00
	FGSM	0.04	0.62	0.96	0.38	0.26	0.29	0.74	0.71	0.03	0.81	0.97	0.19
Outside	CW	0.00	0.01	1.00	0.99	0.01	0.00	0.99	1.00	0.00	0.00	1.00	1.00
	JSMA	0	0.72	1.00	0.28	0.01	0.26	0.99	0.74	0.00	0.06	1.00	0.94
	FGSM	0.00	0.25	1.00	0.75	0.00	0.00	1.00	1.00	0.00	0.00	1.00	1.00
LLM0%	CW	0.04	0.44	0.96	0.56	-	-	-	-	-	-	-	-
	JSMA	0.47	0.50	0.53	0.50	-	-	-	-	0.00	0.81	1.00	0.19
	FGSM	0.39	0.42	0.61	0.58	-	-	-	-	0.00	0.77	1.00	0.23
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

than on vehicle platooning and RUL datasets. Indeed, over the 60% of the JSMA attacks is well detected by both methods, as well as FGSM attacks are by inside

method.It is worth underlying the surprising result for JSMA on DNS, that can be detected very well by using the *inside* method (a plot of the adversarial region is provided in Figure 5.2). As visible from the Figure, such good performance might



(a) Adversarial Region obtained for JSMA attack in DNS tunneling dataset by perturbing the intervals thresholds for features m_A and k_A with *inside* method (TPR=0.93, FPR=0.03, TNR=0.97, FNR=0.07).



(b) Zoom on the part of adversarial region, obtained for JSMA attack in DNS tunneling dataset with *inside* method, corresponding to low values of m_A and k_A

Figure 5.2: Adversarial region with *inside* for JSMA in DNS tunneling

also be partially influenced by the outlier attack data injected for high values of both m_A and k_A. These features represent, respectively, the mean and the kurtosis of the size of the answer packets generated by the DNS server to reply DNS address resolution requests received from clients [5]. By zooming on the lower values as depicted in Figure 5.2b, the portion of attack data contained inside the adversarial region is still acceptable. Nevertheless, the superposition of some attack points on legitimate, observable outside the adversarial region, gives an idea about the difficulty of the problem.
In vehicle platooning dataset, *outside* and *inside* methods are not enough to provide a sufficient performance, indeed, as it can be seen from Table 5.2, TPR does never exceed 0.60.
Regarding the RUL dataset, the inside and outside methods perform in line with the other considered scenarios, the first method overcoming the latter in the detection of all the three adversarial attacks. Despite the geometrical simplicity of the inside method, which looks at 2D adversarial regions by perturbing the thresholds of two features only, it is possible to point out the good balance between FPR and FNR obtained on JSMA attack (an analog result has been obtained for the DNS tunneling too) and FGSM. CW attack remains hardly detectable in this case as well.

The behavior of the two methods, in finding too low TPRs in some cases, may be related to their too simplistic boundaries as defined by the union (with *inside* method) or intersection (for *outside*) of the two intervals considered until now. A way to look for more refined adversarial regions is offered by the third method described in Sec. 4.1, i.e., the *LLM0%* (Eq. 4.7). As the maximum covering was too low in CW and FGSM rules for the vehicle platooning case, the *LLM0%* approach could not be set on this dataset, since it would have required higher covering rules as a starting point. Hence, the *LLM0%* optimization technique was only applied on DNS tunneling and RUL datasets, where the covering percentages were satisfactory in all kinds of adversarial attack, except for the case of CW in RUL dataset. For each attack, the first 5 rules predicting the adversarial class scoring the largest covering were selected and merged in logical `OR` ($\vee$). Then, the thresholds of their conditions related to the first two most important features were perturbed following Eq. 4.7. The performance obtained after the optimization is shown in Table 5.2 . It is interesting to underline that *LLM0%* is the only method that works better on CW attack than on JSMA or FGSM. Differently from the DNS tunneling case, the LLM0% algorithm works very well on JSMA and FGSM attacks for RUL dataset. For the CW attack on DNS, more details are reported about *LLM0%* in the following. The joining in *OR* ($\vee$) of the 5 highest-covering LLM rules with 0% error, before any perturbation, results in the following predictor:

$$\begin{gathered}\textbf{if}(\ (\boldsymbol{m_A > 277.94}) \vee \\ (m_A > 274.55 \wedge 26257 < v_A \leq 39245) \vee \\ (m_A > 271.67 \wedge \boldsymbol{s_{Dt} > 7.61}) \vee \\ (m_A > 269.84 \wedge 8.98 < v_{Dt} \leq 11179 \wedge k_{Dt} > 55.19) \vee \\ (m_{Dt} > 0.95 \wedge m_A > 265.03 \wedge 223.15 < k_Q \leq 543101255))\ \textbf{then attack}\end{gathered}$$

The feature ranking for the attack class indicated features m_A and s_{Dt} being the most relevant. As already mentioned, the first attribute is the average size of the answer packets from the DNS server; s_{Dt} is the skewness of the time interval between queries and answers[5]. Starting by the predictor shown above, the most stringent threshold values corresponding to such features (namely, 277.94 for m_A and 7.61 for s_{Dt}, as previously highlighted in bold) were perturbed. This led us to obtain a new optimized predictor characterized by new thresholds:

$$\begin{gathered}\textbf{if}(\ (\boldsymbol{m_A > 291.83}) \vee \\ (m_A > 274.55 \wedge 26257 < v_A \leq 39245) \vee \\ (m_A > 271.67 \wedge \boldsymbol{s_{Dt} > 8.14}) \vee \\ (m_A > 269.84 \wedge 8.98 < v_{Dt} \leq 11179 \wedge k_{Dt} > 55.19) \vee \\ (m_{Dt} > 0.95 \wedge m_A > 265.03 \wedge 223.15 < k_Q \leq 543101255))\ \textbf{then attack}\end{gathered}$$

Together with the low metrics obtained through inside and outside on CW, this result corroborates the overall difficulty in detecting such an attack through the proposed XAI-based algorithms. Indeed, very intricate overlapping of points in the two classes was observed.

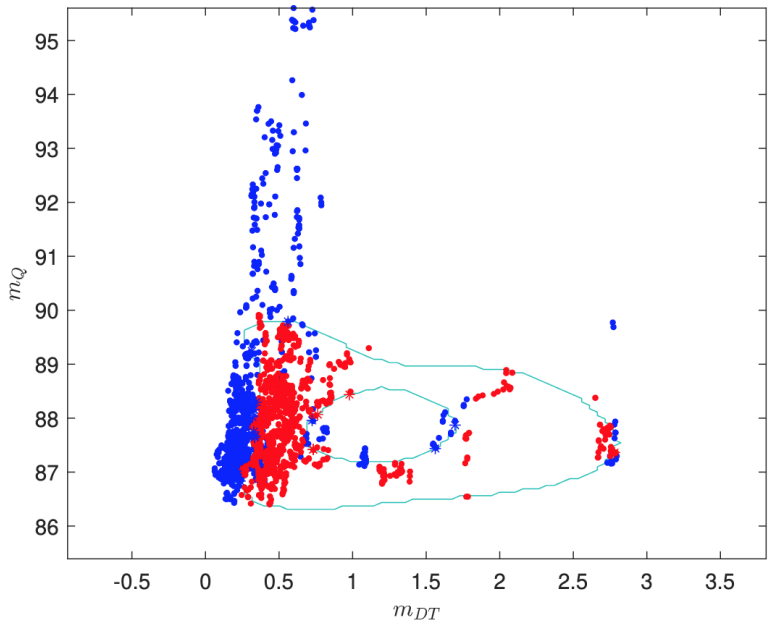


Figure 5.3: 2D graph of the adversarial region (the red points are the attacked ones) with m_{Dt} (average inter-arrival time between query and answer packet over 1000 sample) and m_Q (average size of query packet) as input features of the JSMA attacks on DNS dataset. The star points are the Support Vectors of the description, colored referring to their specific label.

Besides the rule-based reliability methods, a comparison has been carried out by adopting the SVDD-based ones (see the last two rows in Table 5.2). Figure 5.3 shows the shape of the adversarial region obtained on JSMA attack for DNS dataset via zeroFPRSVDD. Although it is far more complex than rules, the overall performance of zeroFPRSVDD on the three datasets resulted in line with the XAI-based methodologies. On the DNS dataset, *zeroFPRSVDD* outperformed (0.03 FPR, 0.22 FNR) all inside, outside and LLM0% in detecting the FGSM attack, while *inside* method revealed better on CW and JSMA. For the vehicle platooning case, *zeroFPRSVDD* better performed on CW (0.11 FPR, 0.79 FNR), whereas the *inside* method was better in detecting JSMA (0.07 FPR, 0.44 FNR) and FGSM attacks (0.26 FPR, 0.71 FNR). Finally, no significant difference was evidenced in RUL dataset. Not surprisingly, the performance of *eXplainableSVDD* is inferior to that of zeroFPRSVDD: with the explainable version of the safe SVDD, complex decision boundaries are approximated with hyper-rectangles, i.e., rules. The explanations provided here are obtained from two steps, i.e., the application of the zeroFPRSVDD and subsequent rule extraction, thus introducing two potential sources of error. From this point of view, although reduced to profiling class boundaries through hyper-rectangles only, native XAI generates the rules in a single

step and may achieve good performance as well. This is corroborated by the reported experiments, where good results were obtained from both native rule-based reliability and explainableSVDD, in terms of balance between FPR and FNR.

5.1.4 Conclusions

This Section presented an innovative approach to detect adversarial machine learning attacks exploiting the rule-based reliable AI techniques described in Section 4.1. Namely, the three algorithms (*reliability from outside*, *reliability from inside* and *LLM0%*) were adopted to search for adversarial regions characterized by the largest portions of attack points while ensuring zero statistical error. Being based on an interpretable model (i.e., the LLM), the results obtained through these methods allow to capture useful knowledge about the adversarial behaviour, even if not always satisfactory in terms of detection performance. Indeed, the adversarial regions are defined in the form of interpretable ranges of values in the space of the considered features and represent a warranty to detect, at least, a small region where the attack infiltrates.

5.2 The Role of Rule Similarity in Wearable Data Augmentation Evaluation

5.2.1 XAI and data augmentation

The development of accurate XAI is often affected by the quality of the available datasets, which can be incomplete (e.g., missing data) or limited due to the complexity of the problem, especially when dealing with human beings. An example is in the healthcare sector, where insufficient populations of patients are monitored with respect to the quantity of available features [111]. Data augmentation has emerged with the aim of building synthetic datasets on the basis of existing data. However, the effectiveness of its adoption in practice is currently under discussion, as there are no closed-form expressions of the probability densities of real versus synthetic data. Statistical validation may drive the augmentation process, but it does not guarantee any property of the quality of the model developed on synthetic data.
In this context, XAI has an important role in evaluating the quality of augmentation processes: for example, it could help identify and mitigate biases introduced into synthetic data, or help shedding light into the data generation process, highlighting and correcting any flaws. However, the adoption of XAI to data augmentation evaluation brings up some challenges in turn: one may indeed criticize on the suitability of the explanations found on the augmented dataset. Investigation on this aspect is then required, starting from the following research questions: are the

new explanations (on the augmented dataset) compliant with what was already known from real data? Are they as accurate as the real ones? Moreover, are the new explanations a valuable piece of information we have actually lost on real data? The last question may be motivated by choosing the right augmentation process, able to expand the data and still preserving their original probability distribution. The augmentation process is however blind to the real probability distribution and should be sustained to discover the most proper data extension, namely, without loosing connection with reality.
This work thus tries to answer these questions, by formulating a validation process of several candidate synthetic explanations, based on rules of the if-then type, together with an innovative rule similarity metric. The resulting methodology defines an automatic process for discovering augmented explanations, still preserving fidelity with original data. Recent research actually addresses the explainability of deep models [15, 124, 6], but without addressing augmentation. After examining the literature on the topic, only [37] was found studying the model variation over real and augmented data, still under neural black-boxes.

5.2.2 Generative Adversarial Networks Evaluation via XAI

In this landscape, rule-based models (Section 2.2) can represent a tool to evaluate the quality of synthetic data generation processes, by exploiting the concept of syntactic rule similarity as introduced in Section 3.2. The overall approach is based on two main pillars: 1) the generation of several synthetic datasets and extraction of their inherent XAI logic (real and fake rulesets in the diagram of Fig. 5.4), and 2) the search of those models that are mostly similar and dissimilar to the original XAI (new knowledge extraction), still preserving fidelity on data (best scenario selection). Being one of the most widespread methods for synthetic data generation, here conditional Generative Adversarial Networks (GANs [51, 90]) are adopted as the reference technique, but this choice does not impact on the method, which remains suitable with any other technique.
Considering classification tasks, two kinds of augmentation-derived rulesets are devised, namely those generated from fake datasets only ("fake XAI model") and from the concatenation of real and artificial datasets ("real+fake XAI model"). These models are used in three ways:

- Scenario 1: train on synthetic, test on real;
- Scenario 2: train on real and synthetic, test on real and synthetic;
- Scenario 3: train on synthetic, test on real and synthetic.

Each scenario involves measuring the quality of the related ruleset performance through two common metrics used in machine learning, i.e. the accuracy and

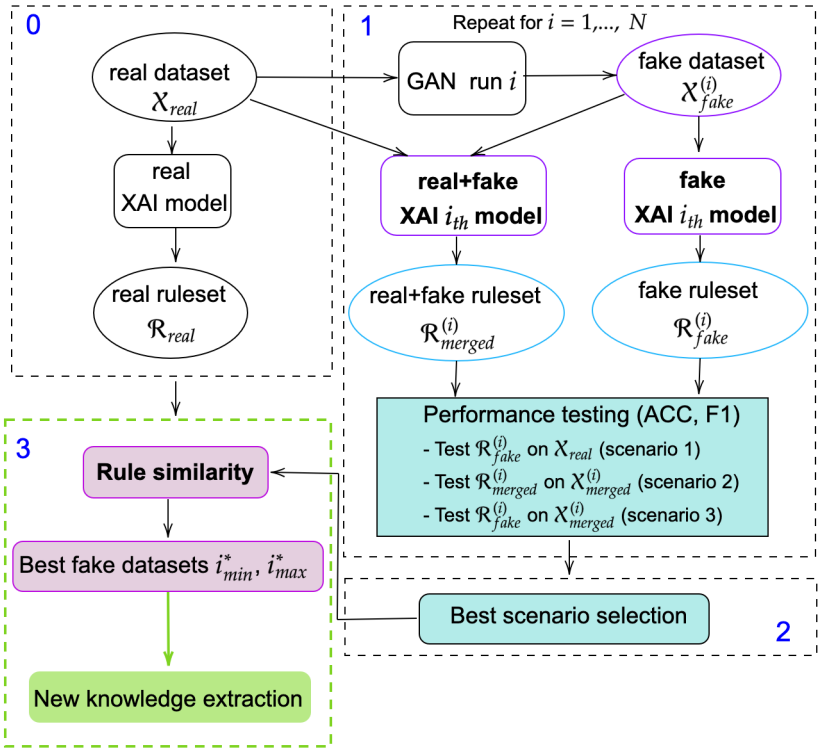


Figure 5.4: High-level idea of the XAI-based GAN evaluation framework. Step 0: knowledge extraction from the real data baseline; step 1: data augmentation and performance assessment; step 2: best scenario selection; step 3: best synthetic datasets selection based on minimum and maximum rule similarity (i^*_{min} and i^*_{max}, respectively), that allow extracting knowledge on the augmentation process.

the F1-score, for each generated dataset. Using this approach, the focus is posed on finding the most appropriate synthetic dataset among those generated by the various augmentation runs. Following the ideas described so far and depicted in Fig. 5.4, a more formal description of the proposed XAI-based evaluation scheme for GANs (in short, XAIGAN) is provided in Algorithm 3.

The first phase (step 0) consists in studying the baseline behavior of the real data, generating the real ruleset $\mathcal{R}_{real}$. Then, step 1 involves the generation of N synthetic datasets and their assessment through the different metrics in each of the described scenarios. Accuracy, F1 score and, for comparison, Frechet Inception Distance (FID) [59] were computed to this purpose. Step 2 performs the selection of the best scenario according to the obtained scores. The best synthetic datasets among

Algorithm 3 XAIGAN

Input: real dataset $\mathcal{X}_{real}$; N: number of GAN runs;

0. **Knowledge extraction from the real baseline**:
a. Train a rule-based classifier on $\mathcal{X}_{real} \rightarrow$ real ruleset $\mathcal{R}_{real}$;

1. **Data augmentation and performance assessment**
For runs $i = 1, \ldots, N$ do:
a. Apply GAN on $\mathcal{X}_{real} \rightarrow$ fake dataset $\mathcal{X}^{(i)}_{fake}$
b. Train a rule-based classifier on $\mathcal{X}^{(i)}_{fake} \rightarrow \mathcal{R}^{(i)}_{fake}$ fake LLM ruleset at run i
c. Merge real $\mathcal{X}_{real}$ with fake data $\mathcal{X}^{(i)}_{fake} \rightarrow \mathcal{X}^{(i)}_{merged}$
d. Train a rule-based classifier on $\mathcal{X}^{(i)}_{merged} \rightarrow \mathcal{R}^{(i)}_{merged}$ real+fake LLM ruleset at run i
e. Get scenario 1 performance: apply $\mathcal{R}^{(i)}_{fake}$ on $\mathcal{X}_{real}$
f. Get scenario 2 performance: apply $\mathcal{R}^{(i)}_{merged}$ on $\mathcal{X}^{(i)}_{merged}$
g. Get scenario 3 performance: apply $\mathcal{R}^{(i)}_{fake}$ on $\mathcal{X}^{(i)}_{merged}$

2. **Best scenario selection** (largest accuracy and F1-score);

3. **Rule similarity → best fake datasets selection → new knowledge extraction**:
a. Minimum similarity
b. Maximum similarity

the generated N are also selected (step 3): this phase exploits the newly introduced syntactic rule similarity metric (Sec. 3.2), that, according to which of the scenarios is selected in previous step 2, involves one of the following comparisons:

- Scenario 1: similarity among rules in $\mathcal{R}_{real}$ and $\mathcal{R}^{(i)}_{fake}$, $\forall i = 1, \ldots, N$;
- Scenario 2: similarity among rules in $\mathcal{R}_{real}$ and $\mathcal{R}^{(i)}_{merged}$, $\forall i = 1, \ldots, N$;
- Scenario 3: similarity among rules in $\mathcal{R}^{(i)}_{merged}$ and $\mathcal{R}^{(i)}_{fake}$, $\forall i = 1, \ldots, N$;

The minimum and maximum rule similarity values will lead to choose two best fake datasets: the first being the one helping discovering new, potentially useful, factors involved in diversifying real and synthetic data, and the latter expressing the shared knowledge between the two, leading to an increased understanding of the data generation approach.

5.2.3 Application to physical activity monitoring

An interesting application to show how XAIGAN algorithm works is the recognition of physical activity performed by a subject based on measurements acquired from wearable IoT devices. Specifically, the PAMAP dataset[103] has been adopted:

it collects data from 9 subjects with different characteristics (age, height, weight, resting heart rate, etc.). In order to consider a personalized approach, also avoiding the noise due to the high inter-subject variability characterizing the dataset, a single subject (subject 109) was considered for the analysis: a 31-year-old male, who was either at *rest* or performing *rope jumping*. 40 features are included, i.e. temperature (T), acceleration (acc), gyroscope (gyr), and magnetometer (mag) data in the 3D space from three Inertial Movement Units (IMUs) located in the hand, chest and ankle.
By following Algorithm 3, the LLM rule-based classifier is trained on the PAMAP dataset to obtain a real baseline of rules $\mathcal{R}_{real}$ discriminating either *rest* or *rope jumping* classes. Although the dataset classes are unbalanced, the generated rules overall perform very well, obtaining high values of accuracy (0.96) and F1 score (0.97). In detail, the following were the rules composing $\mathcal{R}_{real}$:

1. **if** handT $\leq$ 24.90 **then rest** ($C_a = 0.72$)
2. **if** chestmag3 $>$ 37.10 **then rest** ($C = 0.13$)
3. **if** handT $\leq$ 25.15 $\wedge$ handmag2 $\leq$ -32.59 **then rope jumping** ($C = 0.85$)
4. **if** handT $>$ 24.90 $\wedge$ handgyr3 $>$ -1.13 $\wedge$ handmag3 $\leq$ -16.60 $\wedge$ chestmag3 $>$ 18.02 **then rope jumping** ($C = 0.40$)
5. **if** handgyr3 $>$ -0.37 $\wedge$ handmag3 $>$ -29.91 $\wedge$ chestmag1 $>$ 16.94 **then rope jumping** ($C = 0.06$)

The variable concerning the temperature of the subject's hands, $handT$, being included in 3 out of 5 rules, plays a relevant role in distinguishing rest status from rope jumping. Indeed, it stands at the first position in the LLM feature ranking, which also includes $handmag2$, $handmag3$, $chestmag1$, and $chestmag3$ among the most influent variables.
Wearable-based data collection can be limited by improper usage of the devices by the patients and/or by connectivity issues in transferring data, which may result in limited data availability, not sufficient to implement accurate machine learning solutions. This is why synthetic data generation can come into play, providing new datasets large enough to design good predictive models. A conditional GAN (step 1a in Algorithm 3) was then applied to PAMAP subject 109 dataset, with the assumption that the output classes maintain their unbalanced ratio (75% rope jumping, 25% rest). A sensitivity analysis was conducted to choose the best model parameters, especially the number and size of hidden layers in the generator and in the discriminator networks, the number of epochs, the batch size and the learning rate. $N = 10$ runs of GAN training were executed, each associated to a different combination of such parameters. Ten synthetic datasets were generated, scoring a range of FID varying between 1919 and 3325 (the lower the better), and accuracy

Table 5.3: GAN evaluation results through LLM in PAMAP dataset. Accuracy (ACC) and F1-score (F1) are computed for each run of data augmentation in the three scenarios.

	Scenario 1		**Scenario 2**		**Scenario 3**		
Run	**ACC**	**F1**	**ACC**	**F1**	**ACC**	**F1**	**FID**
0	0.55	0.43	0.67	0.54	0.51	0.56	3088.11
1	0.63	0.54	0.83	0.82	0.74	0.71	3325.37
2	0.74	0.79	0.88	0.87	0.85	0.88	2634.05
3	0.81	0.82	0.91	0.93	0.90	0.89	2916.62
4	0.78	0.83	0.89	0.91	0.87	0.88	2611.23
5	0.82	0.74	0.90	0.88	0.88	0.84	2705.52
6	0.80	0.76	0.89	0.87	0.85	0.81	2339.84
7	**0.89**	**0.84**	**0.98**	**0.98**	**0.94**	**0.93**	**1919.30**
8	0.82	0.81	0.96	0.96	0.92	0.92	2025.46
9	0.86	0.83	0.92	0.91	0.90	0.89	1966.35

and F1 score through the rule-based classifier in the different scenarios (see Sec. 5.2.2), as reported below in Table 5.3.
By observing the trend of the accuracy and F1 score across the runs, it can be observed that the scenario leading to the overall largest metrics values was scenario 2, which implied a full augmentation of the real data with synthetic ones before generating new rules, i.e., working with merged datasets $\mathcal{X}^{(i)}_{merged}$, with i being the index of the run.

Rule similarity and new knowledge extraction In the setting of scenario 2, the LLM algorithm trained on the best-performing run $\mathcal{X}^{(7)}_{merged}$, resulted in the following ruleset $\mathcal{R}^{(7)}_{merged}$:

1. **if** handT $\leq$ 25.12 $\wedge$ handacc1 $\leq$ 12.06 $\wedge$ handmag1 $>$ 32.17 **then** **rest** ($C = 0.84$)
2. **if** handmag3 $\leq$ -26.06 $\wedge$ chestT $\leq$ 32.06 **then** **rest** ($C = 0.63$)
3. **if** handacc1 $\leq$ 7.54 $\wedge$ handacc1bis $\leq$ 9.15 $\wedge$ handmag3 $>$ -50.30 $\wedge$ chestgyr2 $\leq$ 1.65 $\wedge$ ankleacc2bis $\leq$ 22 $\wedge$ anklegyr3 $>$ -3.05 $\wedge$ anklemag3 $\leq$ 6.05 **then** **rest** ($C = 0.45$)
4. **if** handT $>$ 24.87 $\wedge$ handmag1 $\leq$ 32.90 $\wedge$ chestmag3 $>$ 12.42 **then** **rope jumping** ($C = 0.86$)

5. **if** handgyr2 $\leq$ 2.71 $\wedge$ handmag3 $>$ -28.93 $\wedge$ ankleacc2bis $>$ -1.71 $\wedge$ ankleacc3bis $>$ -2.63 **then rope jumping** ($C = 0.52$)

6. **if** handT $>$ 25.11 $\wedge$ ankleacc2 $>$ -1.24 **then rope jumping** ($C = 0.45$)

7. **if** handgyr1 $\leq$ -1.913510 $\wedge$ handgyr2 $\leq$ -0.31 **then rope jumping** ($C = 0.08$)

By qualitatively comparing $\mathcal{R}^{(7)}_{merged}$ and $\mathcal{R}_{real}$ some similarities but also some differences emerge. As for the real ruleset, handT is a main feature in the augmented dataset too; in $\mathcal{R}^{(7)}_{merged}$, handmag1 and handmag3 are recurrent, and are also highly correlated with the handmag2 appearing in $\mathcal{R}_{real}$ (all three refer to hand magnetometer data, just differing for the spatial direction). Conversely, $\mathcal{R}^{(7)}_{merged}$ includes variables concerning the ankle as ankleacc2bis, ankleacc3bis and anklemag3, not considered in the $\mathcal{R}_{real}$. In order to capture these aspects quantitatively, rule similarity according to Equation 3.4 was computed between rules from $\mathcal{R}_{real}$ and from $\mathcal{R}^{(7)}_{merged}$; results are reported in Table 5.4, for rules scoring non-zero similarity.

Table 5.4: Rule similarities $q_{\text{synt}}(\cdot,\cdot)$ between each rule in $\mathcal{R}^{(7)}_{merged}$ and in $\mathcal{R}_{real}$.

Rule ID in $\mathcal{R}^{(7)}_{merged}$	**Rule ID in $\mathcal{R}_{real}$**	$q_{\text{synt}}(\cdot,\cdot)$
2	1	0.35
5	3	**0.04**
7	3	0.11
5	4	0.37
7	4	0.10
6	4	**0.51**
7	5	0.32

The highest similarity is between rule 6 in $\mathcal{R}^{(7)}_{merged}$ and rule 4 in $\mathcal{R}_{real}$, since in both rules the variable *handT* appears with a similar threshold. Conversely, some similarities between rules that do not share features are equal to zero, for instance between rule 1 or 3 in $\mathcal{R}^{(7)}_{merged}$ and rule 2 or 1 in $\mathcal{R}_{real}$ (not shown in Table 5.4). The similarity between $\mathcal{R}^{(7)}_{merged}$ and $\mathcal{R}_{real}$ (Eq. 3.5) is $\bar{q}_{\text{synt}}(\mathcal{R}_{real}, \mathcal{R}^{(7)}_{merged}) = 0.24$, which is not particularly high. In fact, the $\mathcal{X}^{(7)}_{\text{merged}}$ dataset's distribution resembles of the distribution of the real dataset (as per the good FID values from table 5.3), also generating new information.

5.2.4 Conclusions

This work proposed an explainable method, based on a rule-based XAI algorithm and an innovative rule similarity measure, to understand and evaluate data augmentation processes (based, in this case, on GANs) and extract knowledge from them. The main innovation relies in the way the proposed approach extends the typical usage of synthetic data, which is to extend real data and to outperform baseline results, by individuating the best synthetic rules (and, in turn, the best synthetic datasets) that, while having a good classification performance, also express new knowledge that can improve the understanding of the considered problem. Also, the proposed framework is flexible with respect to both the kind of augmentation technique (not only GANs) and the rule generation method (not only the LLM).
In the experimentation carried out on the PAMAP use case for activity classification, different sets of rules based on both real and synthetic datasets ($\mathcal{R}_{\text{real}}$ and $\mathcal{R}^{(7)}_{\text{merged}}$) were compared to investigate similarities and differencies.
Besides offering an innovative way to evaluate the results of synthetic data generation processes, this method also represents an effort to combine data augmentation with eXplainable AI, which is fundamental for the adoption of synthetic data in real-world clinical practice, improving the understanding of data and of how augmenting them impacts on the knowledge inference process.
Future works on the topic may include several improvements in different phases of the pipeline, starting by GANs training, with further testing to reach better parameters or improvements to the architecture to accelerate the process. Also, a more complete testing on other wearable devices and clinical data would represent an interesting extension too, as well as the application of the proposed methodology to other fields like automotive, cybersecurity, or aerospace.

5.3 Safety Regions for Robotic Navigation

5.3.1 Background

Robotics is a field where explainability and reliability of AI are fundamental. Ensuring a safe and effective robotic navigation is one of the most challenging tasks though, since trivial solutions in the direction of providing safety guarantees, like drastic navigation speed reductions to avoid collisions among robotic agents, often results in system inefficiencies and deadlock conditions. Reliability then involves finding the best compromise between the safety guarantee and the operating conditions imposed by such guarantees. The problem of ensuring reliable robotic behaviors is being addressed from both the more traditional model-based approaches of control theory and the more innovative data-driven methods raised with the advances in machine learning (especially reinforcement learning, RL): the combination of the two offers a great chance to design safety regions characterized by

strong guarantees of the model-driven component and the generalization ability to unseen contexts of the RL part [21]. However, the potentialities offered by hybrid solutions between the two worlds are still object of investigation in the research community [29]. Traditionally, the assessment of the reliability of these systems exploits classical model-based control techniques from the field of Model Predictive Control (MPC) [9], or advanced emergency breaking systems [8]. Optimal reliability modelling has a long tradition in cyber-physical systems too [77], with the goal of discovering the probability of survival under components failure rate, by expanding the effect of a single component to the system.
In data-driven scenarios, canonical model-based solutions are not always possible and reliability can only be guaranteed by data themselves: in this context, ML provides an estimation of the mapping between the operative domain and the system behavior, by just collecting data samples of the system. The necessity of having reliable systems to assess safety in autonomous system is thus essential in modern ML. The mere prediction optimized to target statistical metrics like accuracy, precision or recall is not anymore sufficient and the disclaim between a good ML model and a bad one is nowadays based on the level of trustworthiness that the system can guarantee. The need of models that can handle with this idea is then more and more required and so is the effort asked to the scientific community. In this sense, the scientific field to be investigated is the so-called "uncertainty quantification", strongly leveraging on statistical learning theory to provide probabilistic assurance to AI models [142, 154]. In this way, also thanks to the development of powerful simulation tools providing the required quantity of data, pure data-driven methodologies can become valuable alternatives on the way to AI certification.

Contribution

A simulated scenario of human-inspired mobile robots navigation is here considered, where robotic agents may collide each other while moving between pairs of opposing targets. In this setting, the exclusive collision avoidance might end up in bringing the agents stuck in their positions, i.e., reaching deadlocks. This research aims at characterizing these two opposite phenomena, collisions and deadlocks, as well as their combination, from the point of view of some crucial parameters of the navigation system. Specifically, both native rule-based XAI alone and black-box reliable AI solutions with rule-based post-hoc explainability are adopted.

5.3.2 Simulation of Social Robotics Navigation

This work is based on simulation data acquired through the social navigation simulator Navground[1], that allows to experiment with robotics navigation algorithms.

[1] **Nav**igation Play**ground**, see https://idsia-robotics.github.io/navground/

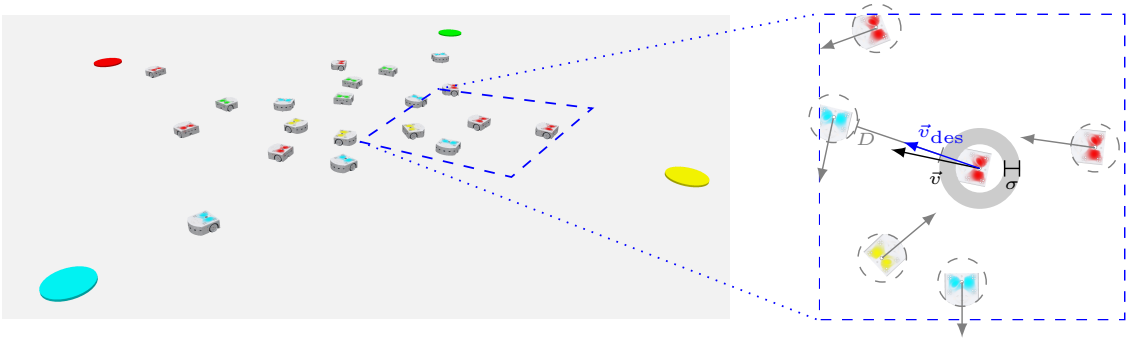


Figure 5.5: Left: the simulated Cross scenario where agents navigate back and forth between pairs of 4 opposing targets, distanced 4 m from each other; the colors of the agents denote their current targets. Right: the navigation behavior for a robot (modelled as a disc with a safety margin σ added to its radius) moving towards the red target: after selecting the desired direction considering its target and the state of its neighbors (modelled as discs), it computes the desired velocity $\vec{v}_{\text{des}}$ taking into account the free distance D in that direction, and then modulates current velocity $\vec{v}$ towards $\vec{v}_{\text{des}}$.

The simulator features multi-agent systems that perform a given navigation task, avoiding collisions with static obstacles and other agents. Each agent is modelled as a disc with a state given by 2D pose and velocity, and navigates using one of the several possible *reactive navigation behaviors* that take the current state of the environment into account to output a control command to progress towards the target while avoiding collisions.
In this work, the Cross scenario is considered, as shown in Fig. 5.5, where four targets are located at the vertices of a cross. One half of the agents navigate back and forth between the red/yellow targets pair, the other half between the green/cyan pair, and the two groups cross in the middle. Each agent is governed by a Human-Like navigation behavior.
The Human-Like behavior (HL, [56]) is a bio-inspired local navigation algorithm that extends and adapts to robotics a heuristic model for pedestrian motion [95]. The behavior tries to address both engineering (e.g., effectiveness of trajectories or scalability to differently crowded environments) and societal aspects, i.e., producing acceptable, human-friendly and predictable trajectories. As illustrated in the right side of Fig. 5.5, the navigation behavior, at regular time steps Δt, performs the following actions to control the agent's velocity:

1. It picks a desired direction where it would come nearest to the target point before possibly colliding with any obstacle or neighbor. For this, the agent enlarges its radius by a *safety margin* σ and takes into account the current velocity of neighbors too.

2. It selects a maximal desired speed that would allow to stop in less than η time: $|\vec{v}_{\text{des}}| = \min(v_{\text{opt}}, D/\eta)$, where D is the free distance in the desired direction and v_{opt} the optimal speed.

3. It modulates the velocity over relaxation time τ: $\dot{v} = \frac{\vec{v}_{\text{des}} - \vec{v}}{\tau}$

Table 5.5: Parameters of the HL behavior considered in the analysis.

Parameter	Description
τ	Relaxation time controlling the smoothness of the motion
η	Minimal time to anticipate unexpected collisions
σ	Minimal distance to keep away from obstacles and neighbors

Parameters impact the safety of the resulting trajectories. For instance, safety margin σ is added to account for modelling and perception errors to reduce the probability of collisions. For the scenario used in this work, where simulated agents have ideal perception, a value $\bar{\sigma}$ can be defined, ensuring that no collision occurs; for differential-drive robots, it is given by

$$\bar{\sigma} = 2v_{\text{opt}}(\Delta t + \tau + \tau_{\text{rot}}) \tag{5.1}$$

where τ_{rot} is an extra relaxation-time for rotations. This value represents a *very conservative* upper-bound because it models a worst-case that happens very rarely if ever. In the next section, a more useful description for the parameters region linked to safe trajectories is derived. Table 5.5 summarizes the HL parameters that are going to investigated from XAI and RAI point of view.

5.3.3 Proposed method

Problem setting

In order to study the robots' social navigation in a comprehensive way, two different objectives have to be pursued: safety and efficiency. The first refers to a navigation behavior that avoids causing collisions with the surrounding agents, while the latter ensures that the robotic motion is still adequate and socially acceptable. For this reason, this study involves three different simulation tasks:

1. collision avoidance, which only refers to *safety*;

2. deadlock avoidance, only accounting for *efficiency*;

3. the ideal scenario of collision and deadlock avoidance, guaranteeing both *safety and efficiency*.

Using Navground tool, a dataset suitable to collect collisions and/or deadlocks at variations of HL behavior parameters has been generated. Specifically, for each simulation run, a 3-dimensional feature vector

$$\boldsymbol{x} = (\tau, \eta, \sigma)$$

was created collecting the parameters of the behavior.
The output target class is determined by measuring the number of collisions, deadlocks or both [2] occurred among all agents. These numbers allow us to define three binary labels y for the simulation runs in the following way:

$$y = \begin{cases} +1 & \text{if number of collisions/deadlock/both } = 0 \\ -1 & \text{if number of collisions/deadlock/both } > 0 \end{cases} \tag{5.2}$$

XAI and RAI are then investigated through the techniques described in the next paragraphs. As shown in Figure 5.6, two different solutions are devised: on the one hand, native XAI solution provides interpretable outcomes but does not guarantee any bound on the classification error; on the other hand, the black-box solution via adjustable classifiers combined with probabilistic and conformal safety regions is designed to provide probabilistic assurance on the error. However, this latter solution lacks of explainability, and post-hoc rule extraction is therefore investigated to locally extract rules from the black-box safety regions boundaries, thus achieving a compromise solution, providing explainability again while maintaining a sufficient level of reliability.

Native XAI solution. For each task, LLM and skope-rules rule-based classifiers are adopted and compared to provide intrinsically interpretable rules that map the inputs with the outcome class.

Probabilistic and conformal safety regions with rule extraction. Native XAI alone does not provide any reliability guarantees itself. Therefore, this part of the work is devoted to first individuate data-driven safety regions with controlled error by adopting black-box methods, based on the notion of adjustable classifiers and probabilistic scaling and a special re-definition of conformal prediction based on it.

[2] These numbers are also provided by the simulator

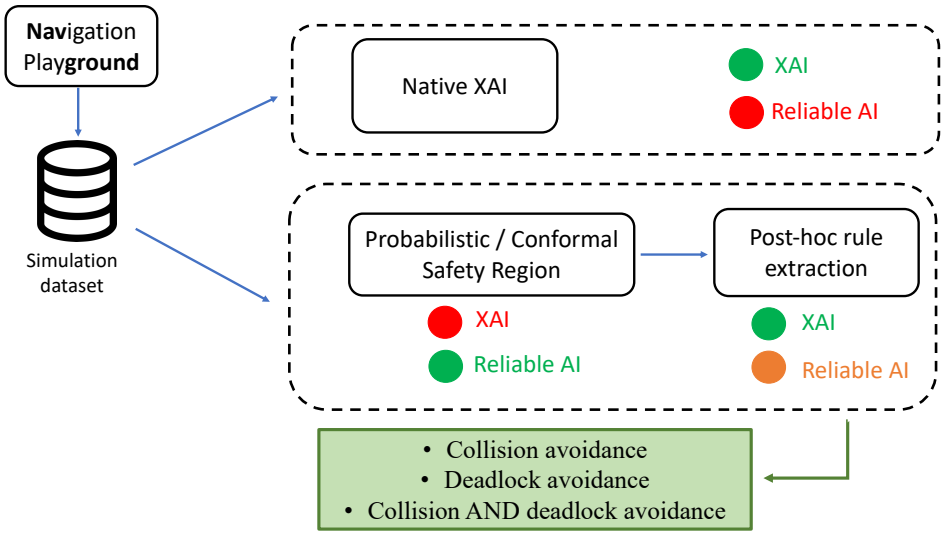


Figure 5.6: Flowchart of the proposed methodology for the design of interpretable safety regions for safe and/or efficient robotics navigation, involving XAI and reliable AI components. Green, orange and red circles are used to denote a good, medium or bad level of the related component.

The starting point is the idea that any classifier, $\hat{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ depending on hyperparameters $\boldsymbol{\theta}$, can be made *adjustable* through the addition of a scalar parameter $\rho \in \mathbb{R}$:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = \hat{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \rho \tag{5.3}$$

A simple example is given by SVM classifier. Its decision function is $\hat{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{w}^\top \varphi(\boldsymbol{x}) - b$, where $\boldsymbol{w}$ is the vector of the weights to be estimated from the data, $\varphi(\cdot)$ is a nonlinear mapping, b is the offset computed on the basis of the support vectors. Its adjustable version, i.e., the *adjustable SVM* is easily obtained adding ρ additively, i.e. $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = \boldsymbol{w}^\top \varphi(\boldsymbol{x}) - b + \rho$.
Given the idea that the class $+1$ refers to a "target" situation (e.g., absence of collision) and -1 to a "non-target" (e.g., presence of collision), the introduction of adjustable classifiers allows to define other two important concepts to assess the reliability of classification, that is the *probabilistic safety regions (PSR)* and the *conformal safety regions (CSR)*. These approaches combine adjustable classifiers notion with probabilistic scaling [87] and conformal prediction (Sec. 4.2.1) theories, respectively. Although in different ways, both consist in individuating the optimal value of ρ, i.e., the optimal adjustment of the black-box classification boundary that keeps the misclassification error at most at a specified level $\varepsilon \in (0,1)$. In both cases, such an adjusted boundary makes it possible to individuate safety regions $\mathcal{S}_\varepsilon$

in the feature space such that:

$$\Pr\{\Pr\{y = -1 \ \wedge \ \boldsymbol{x} \in \mathcal{S}_\varepsilon\} \leq \varepsilon\} \geq 1 - \delta, \tag{5.4}$$

holds for any user-defined error level $\varepsilon \in (0,1)$ and being $\delta \in (0,1)$.
In the following, the notation $\mathcal{S}_\varepsilon^{PS}$ will be adopted to express the safety region obtained with probabilistic scaling approach. Similarly, $\mathcal{S}_\varepsilon^{CP}$ is used for the conformal prediction approach. Since the development of these methods was not part of my personal contribution to this research, I refer the reader to papers [27] and [26, 24] for more formal and in-depth information and theoretical proofs on PSR and CSR, respectively.
In this context, my focus was on the interpretability of these regions instead, which was carried out by using local post-hoc rules through Anchors method (see Sec. 2.3.1). This method was preferred to global rule-based classifiers, since local rules can be built right in correspondence of points of interest near the safety regions decision boundaries. As for all local post-hoc explanation methods (see Sec. 2.3), Anchors requires to select the input instances for which explanations are seeked. In this work, since the objective is to find interpretations for the PSRs and CSRs, these instances are picked within the target class $+1$ and at a short distance $d \leq 0.05$ from the regions decision surface.

5.3.4 Experiments and Results

This Section reports the experimental simulation settings and the results of proposed solutions in the three tasks.

Settings

A total number of 10000 simulation runs were executed, each lasting 5 minutes in simulated time (i.e., a total time of more than one month), with a group of 20 agents modelled after the Thymio robot[3],a small mobile robot with a size of 8 cm and a two-wheel differential-drive kinematics, which is a very common kinematics shared by many ground robots and most smart wheelchairs. Each simulated robot executes the HL navigation behavior with the following parameters: i) $\Delta t = 0.1\,\text{s}$; ii) $v_{\text{opt}} = 0.12\,\text{m/s}$; iii) $\tau_{\text{rot}} = 0.5\,\text{s}$; iv) σ sampled uniformly from $[0.0\,\text{m}, 0.25\,\text{m}]$; v) τ and η sampled uniformly from $[0.0\,\text{s}, 1.0\,\text{s}]$.

Collision avoidance

Figure 5.7, along the top-left/bottom-right diagonal, shows the marginal class distributions of the three features, while the other plots are the features pairwise

[3] https://www.thymio.org

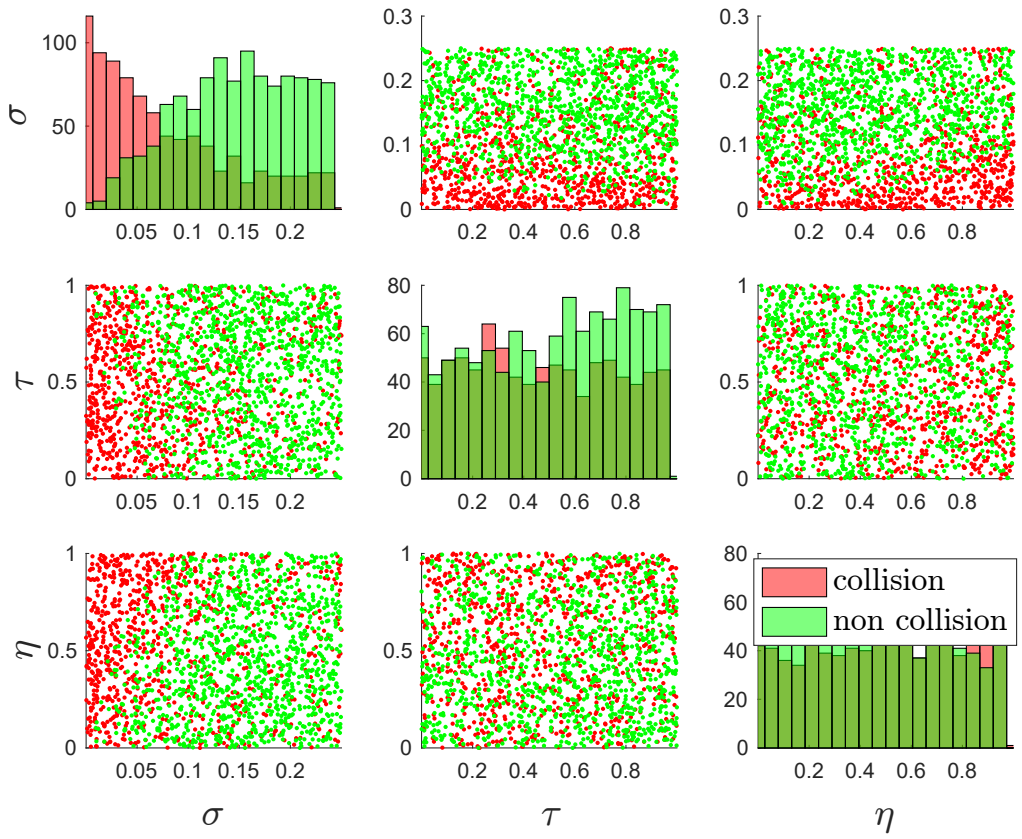


Figure 5.7: Pairwise class distributions of the features in the collision avoidance task.

scatter plots.

Native XAI. Table 5.6 reports the global performance metrics obtained from both models. The first and main difference that shows up resides in the number of rules that were generated, which was more than 4 times higher with the LLM model with respect to skope-rules. This is probably due to the semantic deduplication process carried out in skope-rules algorithm, which filters out rules sharing the same kind of information. Models with less rules have the advantage of being more interpretable, but the richest ones may generate more fine-grained rules with better discriminative ability. And this is what emerges from the higher values of accuracy and F_1-score were obtained through the LLM model, suggesting a better general ability in distinguishing the classes, with good balance between false positives and false negatives.
The labelling criterion (Eq. 5.2) assumes the non collision (+1) class as the positive one, therefore true positives (and therefore the rate TPR) are here referred to non collisions being correctly predicted by the algorithms, while true negatives are

collisions being well classified. Keeping this in mind, since the focus is on trying to best describe the non collision class, the larger TPR reached with skope-rules denotes a better performance in this direction, but with more discrepancy between FPR and FNR.

Table 5.6: **Collision Avoidance.** Performance comparison between the adopted rule-based models. The first column reports the number of generated rules. The other columns refer to the following metrics (expressed in %): accuracy (ACC), F_1-score (F1), true positive rate (TPR), false positive rate (FPR), false negative rate (FNR), true negative rate (TNR).

	# of rules	**ACC**	**F1**	**TPR**	**FPR**	**FNR**	**TNR**
LLM	35	86.7	87.6	89.7	16.6	10.3	83.3
skope-rules	8	83.6	81.9	92.0	27.5	8.0	72.5

Examples of rules predicting *non collision* class are reported in Table 5.7 for both LLM and skope-rules models. These rules were selected, among the others, as they scored the highest covering on test set data, which is an indication of their good generalizability.

Table 5.7: **Collision Avoidance.** Top 3 rules by highest covering on test data, generated via LLM and skope-rules models and predicting the *non collision* class. For each rule, percentage covering and error are measured.

Model	**Top-3 covering rules**	**Covering**	**Error**
LLM	**if** $\sigma > 0.07 \wedge \tau \leq 0.79$ **then** *non collision*	40	4.0
	if $\sigma > 0.019 \wedge 0.096 \leq \tau \leq 0.35$ **then** *non collision*	39	4.5
	if $\sigma > 0.07 \wedge \eta > 0.37$ **then** *non collision*	35	1.8
skope-rules	**if** $\sigma > 0.03 \wedge \eta > 0.25 \wedge \tau \leq 0.63$ **then** *non collision*	51	1.8
	if $\sigma > 0.057 \wedge \tau > 0.59$ **then** *non collision*	21	13
	if $\sigma > 0.03 \wedge \eta \leq 0.36 \wedge \tau \leq 0.63$ **then** *non collision*	35	1.8

The knowledge expressed by these rules confirms the visual information of Fig. 5.7. Apart from little differences in the specific cut-off values, both models agree in

the general shape of non collision class, which can be described by σ over a value ranging between 0.03-0.07m, τ smaller than a value around 0.63-0.79s, and higher values of η.
The qualitative knowledge that these results bring out is intuitive, since it is reasonable that collisions can be avoided by keeping larger distances to the obstacles (higher safety margin). Also, smaller relaxation time τ increases reactivity, leading to more agile maneuvers to avoid collisions, and larger η produces a more careful behavior, reducing speed nearby obstacles and thus collisions too. Compared to the modelled value for the required safety margin, the rules provide a less conservative estimation: for example, $\sigma > 0.07\,\text{m}$ for $\tau \leq 0.79\,\text{s}$, instead of the modelled $0.34\,\text{m}$. As shown, XAI provides a fundamental tool in determining specific cut-off values on these parameters by learning rules' thresholds, which are not known exactly even by field experts.
However, the analysis carried out so far involved standard rule-based classification, and no confidence guarantees were considered. Next step will be therefore dedicated to individuate *safety regions* where non collision class is predicted in high probability.

Post-hoc rule extraction. Anchor rule extraction for the collision avoidance task converged to the same four rules with both PSR and CSR methods. These rules are detailed in Table 5.8, along with an evaluation of their covering and error metrics measured both with respect to the labels assigned via the scaling methods (either probabilistic or conformal) and the ground truth labels. In this way, it is possible to both assess how much the rules are faithful to the SVM-based safety regions, and also how they perform on the actual navigation problem. The rules

Table 5.8: **Rule extraction from collision avoidance safety regions.** Anchors extracted from the adjustable SVM at $\varepsilon = 0.1$, with probabilistic and conformal methods. Covering and error percentages are reported for anchors being tested with respect to the labels assigned via PS ($\mathcal{S}_\varepsilon^{PS}$ output), CP ($\mathcal{S}_\varepsilon^{CP}$ output) and the real labels (Ground Truth column).

Anchor	$\mathcal{S}_\varepsilon^{PS}$ output		$\mathcal{S}_\varepsilon^{CP}$ output		Ground Truth	
	Covering	Error	Covering	Error	Covering	Error
if $\tau \leq 0.51$ **then** *non collision*	76	30	75	28	67	34
if $\sigma > 0.05$ **then** *non collision*	75	26	73	25	76	19
if $\tau \leq 0.25$ **then** *non collision*	44	7.9	44	5.5	36	13
if $\sigma > 0.07$ **then** *non collision*	51	12	49	11	49	8.6

confirm the visual intuitions from Fig. 5.7, i.e., collisions are avoided by increasing σ and lowering τ. The parameter η is not involved in the generated rules, suggesting its marginal role in profiling the non-collision class in high probability. For all rules, the covering rate is satisfactory, underlining their ability in approximating sufficiently well the safety regions. The error is on average higher than that achieved by the probabilistic and conformal scaling, but this is expected due to the simple shape of rules (hyper-rectangles) with respect to the polynomial boundary of the adjustable SVM. In light of this, the results can be considered as a promising compromise between safety and transparency. Nevertheless, reminding the goal of bounding the false positive rate to 10% (i.e., $\varepsilon = 0.1$), it can be observed that two of the four anchors are close to this bound too. Specifically, these are $\sigma > 0.07$ and $\tau \leq 0.25$. Therefore, the logical union of these two rules was performed and

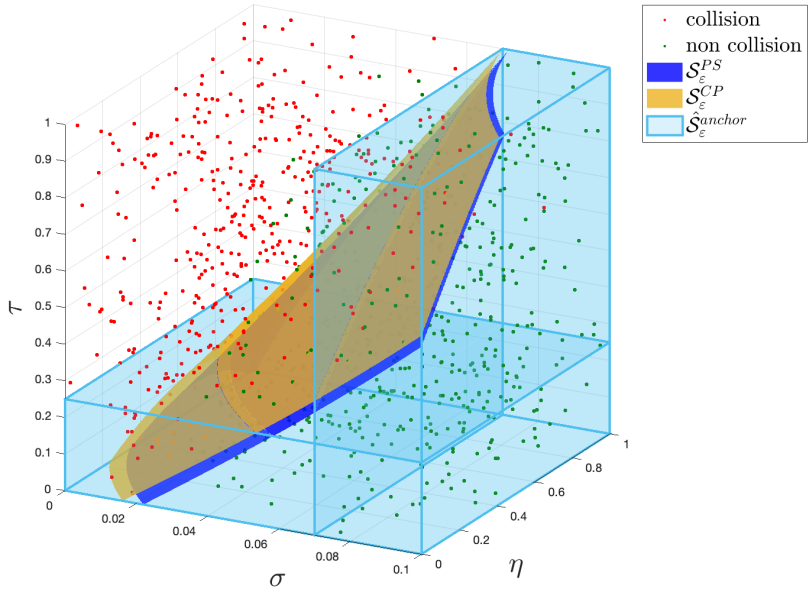


Figure 5.8: **Collision avoidance safety regions.** 3D representation of probabilistic ($\mathcal{S}_\varepsilon^{PS}$) and conformal ($\mathcal{S}_\varepsilon^{CP}$) safety regions, along with the extracted anchors for the collision avoidance task.

evaluated, as synthesised by the following region and represented through the light blue parallelepipeds of Fig. 5.8:

$$\hat{\mathcal{S}}_\varepsilon^{anchor} : \textbf{if } \sigma > 0.07 \text{ or } \tau \leq 0.25 \textbf{ then } \textit{non collision}$$

which scores the 70% of covering and 21% of error on the ground truth labels (78% and 19% on the $\mathcal{S}_\varepsilon^{PS}$ labels, 77% and 16% on the $\mathcal{S}_\varepsilon^{CP}$ labels). This region

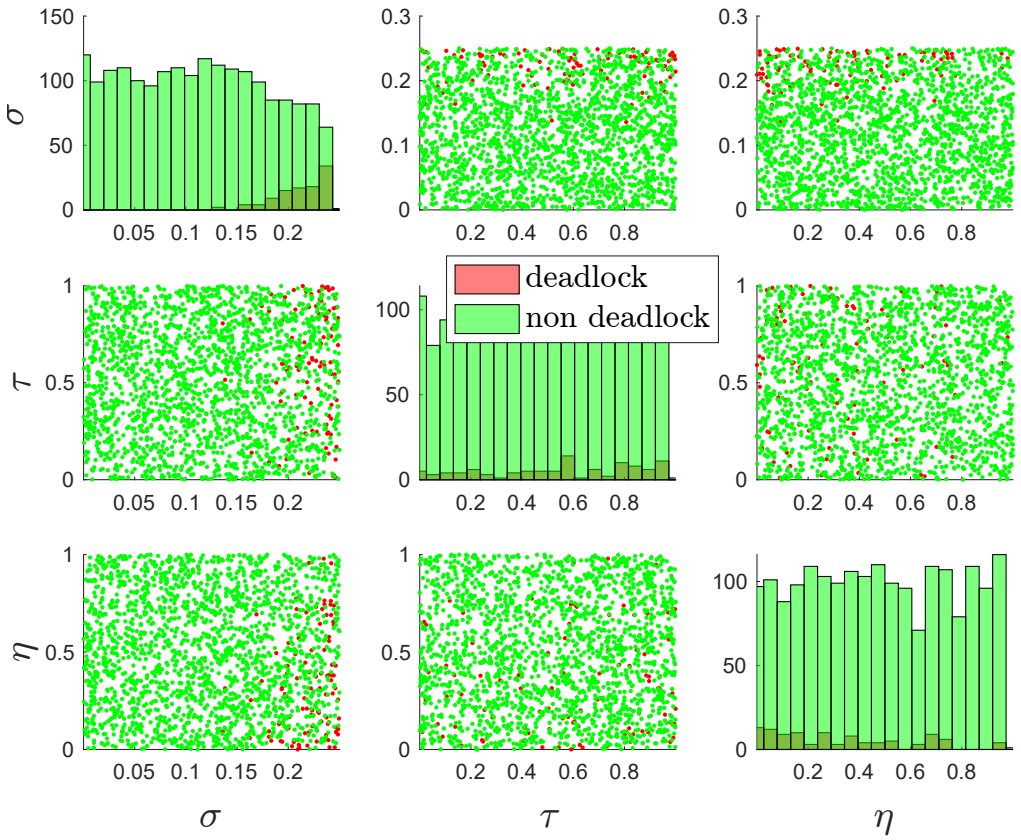


Figure 5.9: **Deadlock avoidance dataset.** Pairwise class distributions of the features in $\mathcal{T}_{nav}$ for the deadlock avoidance dataset.

converts the equations guiding the safe navigation, i.e., those defined by the $\mathcal{S}_{\varepsilon}^{PS}$ and $\mathcal{S}_{\varepsilon}^{CP}$ curves, into simpler recommendations on how the parameters should vary when such safety guarantees are provided. As with native XAI, it is possible to observe how the safety margin threshold $\sigma = 0.07$ m discovered via the Anchors is well below the modelled worst-case value $\bar{\sigma} = 0.38$ m (see Eq. 5.1). This further highlights the importance of devising data-driven approaches.

Deadlock avoidance

The goal of this task is to understand which patterns allow to avoid deadlocks when the autonomous robots move. The pairwise plot in Figure 5.9 illustrates how the two classes (deadlock and non-deadlock) are unbalanced in this case. The situations such that the system goes to deadlock are very few but not zero, so, for safety purposes, is it necessary to individuate the regions of the input parameters where the probability of observing a deadlock is bounded. To apply the reliable AI methodologies as it was done for the collision avoidance problem, it was necessary to increase the size of the "deadlock" class. SMOTE oversampling technique [43]

Table 5.9: **Deadlock avoidance performance.** Performance comparison between the adopted rule-based models. The first column reports the number of generated rules. The other columns refer to the following metrics (expressed in %): accuracy (ACC), F_1-score (F1), true positive rate (TPR), false positive rate (FPR), false negative rate (FNR), true negative rate (TNR).

	# of rules	**ACC**	**F1**	**TPR**	**FPR**	**FNR**	**TNR**
LLM	23	85	84	83	14	16	86
skope-rules	10	83	82.5	67	7.0	33	93

was then adopted to balance datasets.

Native XAI. Table 5.9 shows the obtained performance of the LLM and skope-rules models for the deadlock avoidance task dataset with SMOTE augmentation. It can be seen that, also in this case, the LLM generated more rules (i.e, 23) than skope-rules (i.e, 10). While accuracy and F_1 score are close to those obtained on the collision avoidance task, for both models, the largest differences emerge in skope-rules performance measured through the FPR and FNR metrics. Indeed, this model achieved the 7% FPR with 33% FNR on the deadlock avoidance task, against 27.5% and 8% with collision avoidance. This reflects the better ability of skope-rules in determining deadlock points instead of non deadlock ones. On the other hand, the LLM keeps relatively low errors in deadlock avoidance task, but the FNR worsened with respect to collision avoidance, which highlights an overall higher difficulty in avoiding deadlocks. This is also visible by looking at the three top-covering rules for deadlock avoidance in Table 5.10: just the first rule has a high covering rate, but the other ones score far lower covering (below 14%) and increased error. Looking at the conditions expressed by these rules, it is possible to observe the opposite behavior of the safety margin σ with respect to collision avoidance task, that is, absence of deadlocks occurs when it decreases. Regarding τ and η parameters, the rules showed quite unstable thresholds (i.e., different values through the ruleset), which further highlights the more challenging nature of this task.

Post-hoc rule extraction. Concerning local rule extraction from adjustable SVM-based safety regions, Figure 5.10 reports the plot of the safety regions obtained with the two methods (PS and CP) and their approximation with the Anchor rules. As for the collision task, it can be seen that there is no evident difference

Table 5.10: **Deadlock avoidance rules.** Top 3 rules by highest covering on test data, generated via LLM and skope-rules models and predicting the *non deadlock* class. For each rule, percentage covering and error are measured.

Model	Top-3 covering rules	Covering	Error
LLM	**if** $\sigma \leq 0.174$ **then** *non deadlock*	74.48	3.47
	if $\sigma \leq 0.239 \wedge 0.220 < \eta \leq 0.470 \wedge 0.136 < \tau \leq 0.724$ **then** *non deadlock*	13.95	7.81
	if $\eta \leq 0.112 \wedge \tau > 0.172$ **then** *non deadlock*	11.11	4.68
skope-rules	**if** $\sigma \leq 0.155 \wedge \tau \leq 0.975$ **then** *non deadlock*	64.92	1.21
	if $\sigma \leq 0.155 \wedge \tau > 0.022$ **then** *non deadlock*	64.45	1.32
	if $\sigma \leq 0.144$ **then** *non deadlock*	61.68	0.73

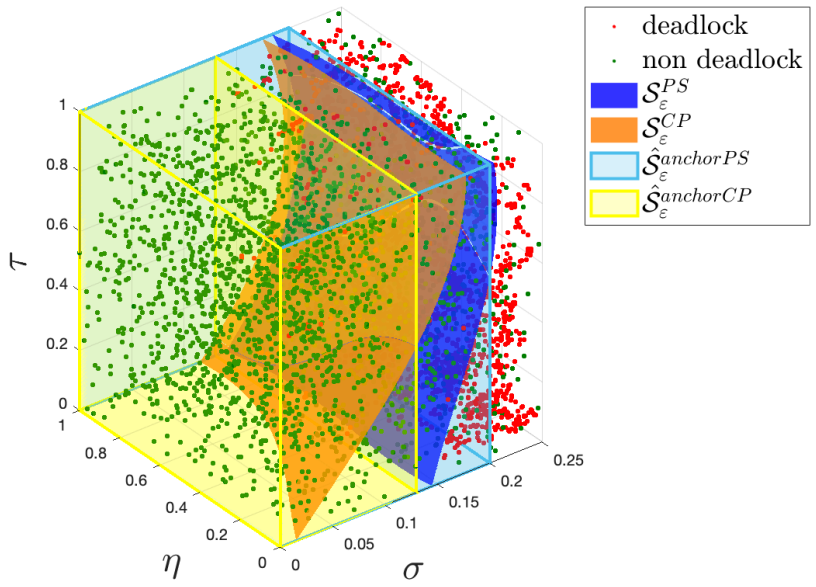


Figure 5.10: **Deadlock avoidance safety regions.** 3D representation of probabilistic and conformal safety regions, with their rule-based approximations via Anchors, in the deadlock avoidance task.

between the two regions and, consequently, the rules extracted from them are approximately similar. By inspecting a selection of the obtained Anchor rules, as reported in Table 5.11, it can be observed, again, that the non deadlock safety regions are characterized by decreasing σ values. Also, slight differences emerged between the anchors extracted from the PSR and those from the CSR: in the latter case, a rule with good covering on true labels shows a role of η variable, while this

Table 5.11: **Rule extraction from deadlock avoidance safety regions.** Anchors extracted from the adjustable SVM at $\varepsilon = 0.1$, with probabilistic and conformal methods. Covering and error percentages are reported for anchors being tested with respect to the labels assigned via PSR or CSR ($\mathcal{S}_\varepsilon$ output column) and the real labels (Ground Truth column).

		$\mathcal{S}_\varepsilon$ output		Ground Truth	
	Anchor	Covering	Error	Covering	Error
PSR	**if** $\sigma \leq 0.20$ **then** *non deadlock*	98	13	84	15
	if $\sigma \leq 0.23 \wedge \tau > 0.71$ **then** *non deadlock*	32	9	27	10
	if $\sigma \leq 0.23 \wedge \tau > 0.44$ **then** *non deadlock*	61	18	53	21
CSR	**if** $\sigma \leq 0.20 \wedge \tau > 0.44$ **then** *non deadlock*	66	6	47	5
	if $\sigma \leq 0.13$ **then** *non deadlock*	78	2	55	0.06
	if $\sigma \leq 0.20 \wedge \eta > 0.55$ **then** *non deadlock*	51	9	37	8

does not happen in the former. While the rules' covering values are on average comparable with the collision avoidance task, the error percentages are here overall lower and, for the CSR technique, they all satisfy the ε bound. One main rule can be identified for each safety region, being the one with the largest covering on the scaling output: these rules are very close to each other, being $\sigma \leq 0.20$ with PSR and $\sigma \leq 0.13$ with CSR. These two rules are depicted with the light blue and yellow parallelepipeds of Fig. 5.10, respectively: the CSR provides the most precise solution, even if having a lower covering. The error associated to it (0.06%) is indeed well below the ε bound, while the anchor extracted from PSR reaches the 15% error (it indeed includes more deadlock samples in its volume).

Collision and deadlock avoidance

In this final task, the aim is to let the robots navigating by following both a non-collision and non-deadlock behavior. Since the original "deadlock avoidance" dataset has very few contributions in terms of unsafe situations, the composition of the dataset is pretty similar to the "collision avoidance" one, as evidenced by the pairwise feature plots of Fig. 5.11.

Native XAI. Table 5.12 reports the obtained global performance metrics for the LLM and skope-rules models on the collision and deadlock avoidance task dataset. The number of rules generated by the LLM (i.e., 52) is more than twice the number of rules found via skope-rules (i.e., 19). Moreover, the LLM registered better accuracy and F1 score than skope-rules (respectively, 85% and 72% against 66% and 70%), although presenting larger errors with 40% FPR and 21% FNR (cfr.

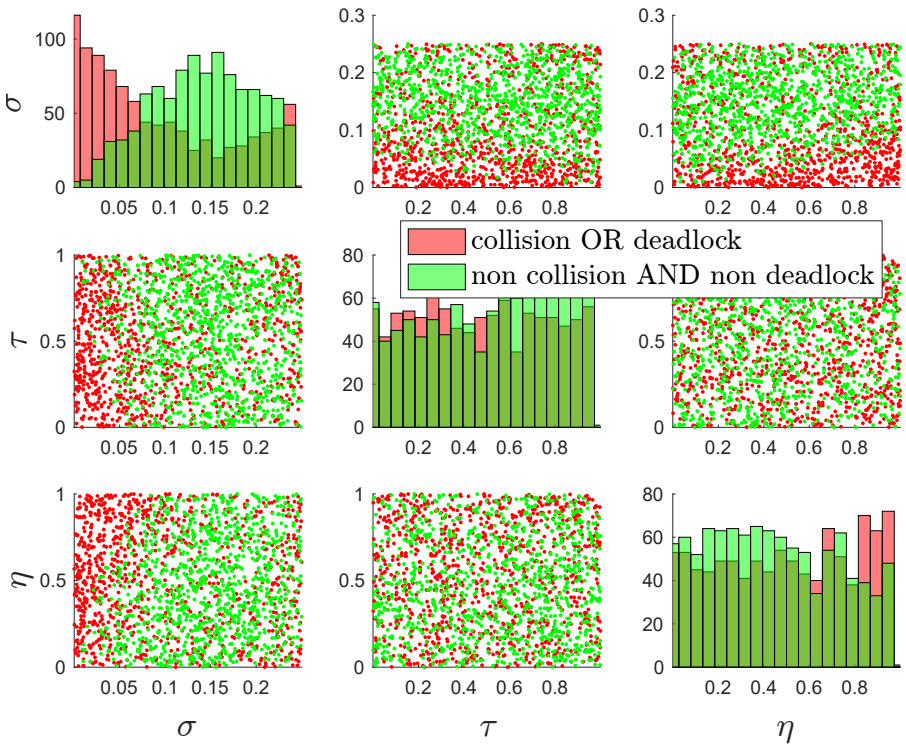


Figure 5.11: **Collision and deadlock avoidance dataset.** Pairwise class distributions of the features in $\mathcal{T}_{nav}$ for the collision and deadlock avoidance dataset.

Table 5.12: **Collision and deadlock avoidance performance.** Performance comparison between the adopted rule-based models. The first column reports the number of generated rules. The other columns refer to the following metrics (expressed in): accuracy (ACC), F_1-score (F1), true positive rate (TPR), false positive rate (FPR), false negative rate (FNR), true negative rate (TNR).

	# of rules	ACC	F1	TPR	FPR	FNR	TNR
LLM	52	85	72	79	40	21	60
skope-rules	19	66	70	95	32	5	68

32% and 5% with skope-rules), highlighting the increased complexity of this task combining collision and deadlock avoidance goals.
This is also confirmed when inspecting the first three top-covering rules reported in Table 5.13: while the LLM keeps the errors under 6%, covering values are below 18%; conversely, skope-rules finds wider rules, with coverings up to approximately

Table 5.13: **Collision and Deadlock avoidance rules.** Top 3 rules by highest covering on test data, generated via LLM and skope-rules models and predicting the *non collision and non deadlock* class. For each rule, percentage covering and error are measured.

Model	Top-3 covering rules	Covering	Error
LLM	**if** $0.158 < \sigma \leq 0.199 \wedge 0.139 < \eta \leq 0.903 \wedge \tau > 0.093$ **then** *non collision and non deadlock*	17.29	5.81
	if $0.029 < \sigma \leq 0.220 \wedge 0.342 < \eta \leq 0.935 \wedge 0.111 < \tau \leq 0.308$ **then** *non collision and non deadlock*	12.10	5.34
	if $0.042 < \sigma \leq 0.224 \wedge 0.799 < \eta \leq 0.895$ **then** *non collision and non deadlock*	11.57	4.05
skope-rules	**if** $0.059 < \sigma \leq 0.224 \wedge \tau \leq 0.951$ **then** *non collision and non deadlock*	80.34	45.03
	if $0.086 < \sigma \leq 0.226$ **then** *non collision and non deadlock*	73.83	37.19
	if $\sigma > 0.065 \wedge \tau > 0.467$ **then** *non collision and non deadlock*	47.27	32.99

80%, but raising the error percentages up to 45%. Furthermore, differently from the two previous tasks, the rules obtained with the LLM delimit the safety margin σ between two extremes and the η parameter appears more frequently, although there is less agreement on the thresholds values, making it less trivial the individuation of the most adequate ones.

Post-hoc rule extraction. Table 5.14 reports a selection of the rules and metrics obtained in the collision and deadlock avoidance case. The covering evaluated on the adjustable SVM output, either for $\mathcal{S}_\varepsilon^{PS}$ and $\mathcal{S}_\varepsilon^{CP}$, is overall satisfactory, even though it is not so high on ground truth labels, which is consistent with the noisy structure of the dataset observed in Fig. 5.11. However, the error with respect to the real labels is sufficiently low, approximately in line with the ε bound.

Table 5.14: **Rule extraction from collision and deadlock avoidance confidence regions.** Anchors extracted from the adjustable SVM at $\varepsilon = 0.1$, with probabilistic and conformal methods. Covering and error percentages are reported for anchors being tested with respect to the labels assigned via PSR or CSR ($\mathcal{S}_\varepsilon$ output column) and the real labels (Ground Truth column).

		$\mathcal{S}_\varepsilon$ output		Ground Truth	
	Anchor	Covering	Error	Covering	Error
PSR	**if** $\sigma > 0.19 \wedge \tau > 0.24 \wedge \eta > 0.23$ **then** *non collision AND non deadlock*	82	4	18	11
	if $\sigma > 0.19 \wedge 0.51 < \tau \leq 0.76 \wedge \eta \leq 0.75$ **then** *non collision AND non deadlock*	25	1	5	2
	if $\sigma > 0.19 \wedge \tau > 0.24 \wedge \eta \leq 0.75$ **then** *non collision AND non deadlock*	72	5	17	10
CSR	**if** $\sigma > 0.19 \wedge \tau > 0.24 \wedge \eta \leq 0.75$ **then** *non collision AND non deadlock*	72	5	17	10
	if $\sigma > 0.19 \wedge \tau > 0.24$ **then** *non collision AND non deadlock*	95	8	23	14
	if $\sigma > 0.19 \wedge \eta > 0.48$ **then** *non collision AND non deadlock*	66	6	16	12

What is noticeable here relies in the knowledge that can be extracted from the

Table 5.15: Performance comparison between probabilistic safety regions and conformal safety regions, in terms of the true positive rate (TPR) in %, i.e., the portions of target points correctly belonging to the safety regions.

	PSR		CSR	
	ρ_ε	TPR	ρ_ε	TPR
Collision Avoidance	0.22	84	**0.14**	**85**
Deadlock Avoidance	**-0.29**	**79**	0.29	64
Collision and Deadlock Avoidance	**1.09**	**16**	1.1	15

obtained rules, which confirm the observations made in the exploration phase of the dataset with respect to the safety margin parameter. That is, safety is here guided through higher values of σ, as it is highlighted in the rules of Table 5.14, and as it was for safe navigation in Table 5.8. In contrast, the efficiency component arises from larger values of τ parameter, coherently with the results in Table 5.11. Moreover, the η parameter emerged more frequently than in the other two cases, suggesting some 'regulatory' role between the two goals (safety and efficiency). Thus, the rules emerged here point out that a safe and efficient navigation can also be viewed as a combination of the two distinct contributions that were observed in the separate study of safety (collision avoidance task) and efficiency (deadlock avoidance task).

5.3.5 Discussion and conclusions

In the three presented experimental tasks, the conditions associated to a safe navigation (collision avoidance), an efficient navigation (deadlock avoidance) and a safe and efficient navigation (collision and deadlock avoidance) were characterized.
Through the adjustable SVM at an error level of $\varepsilon = 0.1$, probabilistic safety regions and conformal safety regions are described by relatively simple equations, that depend on the proper choice of the ρ parameter in Eq. 5.3. Such equations can then be used to guide robots behaviour towards the desired navigation properties (as in Table 5.15). Being similar in the idea, the two approaches (PSR and CSR) can be compared by looking at how many points are correctly covered by each region, i.e. the true positive rate (TPR), which is strictly related to the scaling parameter ρ_ε. To achieve the desired level of error ε, the SVM decision boundary is translated by a quantity defined by ρ_ε in the direction of the target class ($y = +1$) points, thus reducing the size of the region at the increase of the translation entity:

as a result, the best solution between PSR and CSR is the one that performs the smallest variation of the original boundary, i.e., that corresponding to the smallest ρ_ε, thus allowing to include more points in the region. Table 5.15 summarizes the results from this point of view, highlighting conformal prediction-based approach as the best solution for collision avoidance only, moving to probabilistic scaling when it comes to avoiding deadlocks or, more interestingly, both collisions and deadlocks. One may reasonably argue that the nature of these regions is somehow black-box, not providing a clear evidence on the ranges of input parameters they correspond to. This is why rule extraction from such confidence regions has a fundamental role, helping shedding light into such parameters ranges, and increasing the understanding of the collision/deadlock avoidance processes. Explainability is crucial in the comparison between the worst-case and data-driven solution (see the results of the collision avoidance task). However, the desired performance guarantee is hardly achievable through the sharp boundaries of **if-then** decision rules, which can only provide good approximations of the confidence regions. In summary, this work highlighted the cooperative role that can exist between explainability and black-box (probabilistic and conformal) regions, with the first driving knowledge extraction and the second offering support to reliability when the nature of the dataset is too intricate to be tackled with the only XAI.

Chapter 6

Conclusions and Future Work

This Chapter summarizes my doctoral dissertation, emphasizing its achievements, its limitations and possible future developments.
My main PhD research explored a crucial topic in trustworthy AI, that is the intersection between explainability and reliability in machine learning, to which I contributed by taking rule-based binary classifiers as a reference and developing methods to ensure their performance.

- **Rule similarity.** In order to provide rule-based classifiers with a full set of tools that allow to synthesize and compare the syntactic and/or semantic information expressed by a set of rules, I introduced three novel rule similarity metrics: syntactic, Bag of Words, and geometrical rule similarity. Both syntactic and BoW similarity focus on explicit rules conditions only, implying that two rules can have a similarity value larger than zero even if they are not geometrically overlapped. Geometrical rule similarity considers both explicit and implicit rules conditions, thus providing a non-zero similarity only if there is an actual overlap between the rules premises. Each of these metric, though based on a different approach, revealed useful in practical applications, such as: (i) evaluating the quality of synthetic datasets generated via data augmentation (syntactic rule similarity); (ii) profiling subsets of the data according to some variables of interest, also studying how these variables reflect in the output class (BoW similarity); (iii) measuring the geometrical overlap between rules, thus understanding how they are related to each other in the feature space (geometrical rule similarity).

  The first two metric developed so far are both syntactic, since they look at the structure of the rules, while the third is an attempt towards defining a semantic approach. Future research on this topic may thus be devoted to implement new methodological approaches towards an integrated metric combining syntax and semantics. Moreover, new applications of rule similarity

can be investigated, including guiding rule aggregation processes, conducting meta learning analyses, or extracting knowledge about the reasoning of generative AI models.

- **Rules optimization for error control.** I designed three rule optimization methods - reliability from inside, reliability from outside and rules with zero error - for identifying rule-based safety regions with minimized statistical error, starting from different views of the problem. The essence of these methods lies in finding optimal perturbations of rule thresholds themselves, opportunely synthesized by feature and value ranking, through a "grid-search-like" approach that allows to reduce the error on a desired target class. The approaches are heuristic, in that they only rely on the data under analysis and on the information, provided by feature and value ranking, about where, in the feature space, the optimal thresholds should be sought. Optimality is therefore here defined in relation with the empirical performance collected during the exploration of several candidate threshold perturbations, though not formally defined through more theoretical analyses. Nevertheless, this set of methods proved useful in safety-critical applications such as the collision avoidance in vehicle platooning scenarios, the prediction of physical fatigue, and also the detection of adversarial ML attacks.

  A limitation of these approaches is that the grid of threshold candidates is currently defined through the manual setting of a step size. Future research will thus involve optimizing such a candidate generation process, so to reach a faster solution. Also, further testing may be done through cross-validation in presence of a large amount of data, including the adoption of data augmentation techniques. The characterization of the placement of the points deserves further study to understand the optimal covering of the safety regions.

- **Conformal prediction for rule-based models.** Seeking the same objective of identifying rule-based safety regions, I devised a more formal approach relying on conformal prediction theory. CONFIDERAI score function has been introduced to efficiently perform CP in rule-based binary classification, by considering both rules relevance and geometry. Furthermore, on top of CONFIDERAI results, the conformal critical set has been defined, paving the way to the generation of new rules that have improved performance, in terms of precision, on the target class. Extensive experimentation on both toy datasets/rulesets and real-world applications has shown promising results, highlighting the relevance of this contribution in the intersection of explainability and reliability.

  This research is a however just a starting point for the development of a fully conformal rule-based method for trustworthy AI, and some limitations have still to be addressed. One of them regards the overlaps handling: if, on the

one hand, CONFIDERAI effectively works when rules intersect, thanks to geometrical rule similarity, it does not yet properly account for rules "neighborhood" in the case rules do *not* overlap. The score values for points closer to rule boundaries should indeed consider how much a rule is "near" (though not overlapped) to other rules predicting the same or a different label. To this aim, further studies on how to properly quantify such concept of "closeness" are needed in future research: in this respect, investigating the role that the BoW or syntactic rule similarity may play could be a starting point.

Future work will also involve a more in-depth experimentation of other rule-based models and their assessment on real world-applications, as well as the extension of the proposed score function to multi-class problems. Moreover, further investigation could be devoted to better formalize the transition from the original rules to those arising from conformal critical sets.

Appendix A

Other Contributions

This Appendix describes the activities I carried out in parallel with my dissertation research, either as part of research projects (A.1), reliable AI topics of my research group at CNR-IEIIT (A.2) and during my abroad research period (A.3).

A.1 Research Projects

- Advances in **PNEU**mology via ICT and Data Ana**LYTICS** (**PNEULYTICS**): funded by the Italian foundation *Fondazione Compagnia di San Paolo* this project aimed to develop a technological framework, based on medical devices and AI-based analysis tools, for the remote monitoring of patients with respiratory diseases. By combining clinical data with daily life monitoring parameters, the ultimate goal was to provide doctors with a dynamic overview of the disease and its effects on patients' lifestyles, thus enabling a personalized approach to healthcare.

  In this context my contributions mainly regarded the management and preliminary analyses of the project's databases, namely:

  1) **Smartwatch data analysis**. A Fitbit Versa 3 smartwatch was adopted to collect one month of measurements from a subject who was following a pharmaceutical therapy for Chronic Obstructive Pulmonary Disease(COPD). These included heart-rate related measurements, respiratory parameters during sleep, and daily minutes of activity performed at different levels of intensity. Samples were labeled based on the quantity of drug taken by the patient, i.e., 1 puff/day or 2 puffs/day. The built dataset was analysed through the LLM model (2.2.3) to assess which of the considered measurements exhibited the larger variations as the involved patient was treated with a different therapy level. Preliminary testing, and the obtained rules inspection, has provided a few factors as

the most influent in predicting the therapy, namely the average blood saturation and the average heart rate.

2) **Cough-related quality of life.** This study deals with adopting rule-based XAI to support the diagnostic path for identifying the origins of chronic cough-related quality of life (QoL) impairments in asthmatic patients. The analysis is developed around questionnaire data: i) a symptoms questionnaire with self-reported scores related to asthma, pharynx-larynx, upper airways, or gastroesophageal reflux domains; ii) the Asthma Control Test (ACT) [100], measuring the level of asthma control patients have; iii) the Chronic Cough Impact Questionnaire (CCIQ)[14], determining near normal QoL and abnormal QoL classes.The LLM model is adopted to classify patients - opportunely grouped according to their ACT level - based on the symptoms questionnaire, coming up with rules that help discovering which are the key factors affecting the QoL and their corresponding values. Main findings show that these factors include the pharynx-larynx and supper airways when patients have a good control over their asthma, while asthma itself and digestive traits result predominant in patients with worse asthma control. By prioritizing some symptoms categories over other ones, these results may provide clinicians with useful guidance in selecting the most appropriate diagnostic and therapeutic plans.

- **Genova 5G**: funded by the *Italian Ministry of Economic Development* (MiSE), this project explored the potentials of 5G technology in a smart mobility context. The project's use case was a driver alert system for bus, based on video content analysis (VCA) and Message Queuing Telemetry Transport (MQTT) communication over 5G network. The system has been established in a urban area in Genoa, Italy, at an intersection characterized by limited visibility, aiming to detect and signal possible dangerous situations occurring in the area to the bus driver.

  The base VCA system was based on a Single Shot Detector (SSD) [82] model trained on images from a single camera. I studied the possible advantages of using an additional camera. To this end, the base SSD model was first retrained by adding new images from the second camera, and then evaluated in terms of: i) object detection performance, i.e., how well it correctly identified different classes of objects, and ii) alert generation performance, that is the ability of the system to trigger an alert if and only if at least one object is present in the monitored area. The obtained results point out that the addition of the second camera results beneficial, by improving the robustness of the VCA system in case of malfunctions of the first camera, and also leading to a higher number of correctly detected alarms thanks to a wider coverage of the surveilled area.

- **RE**liable & e**X**plainable **S**warm **I**ntelligence for **P**eople with **R**educed m**O**bility (**REXASI-PRO**): funded by the *European Commission* in the Horizon Europe program, this project aims to release a novel framework, based on Trustworthy Artificial Intelligence principles, for social navigation of smart wheelchairs [88]. The main use case involves setting up a collaborative indoor navigation environment constituted by the autonomous wheelchair, that moves to reach a user-selected target destination, a flying robot creating a map of the environment and signaling possible temporary obstacles, cameras to detect predetermined types of events, and an orchestrator computing the best global path for the wheelchair based on information collected from the drone and the cameras.

  In this project, I have been mainly involved in two activities:

  1) Developing methods for the explainability and reliability of the simulation-based robotic navigation, to ensure collision and/or deadlock free movement. Since this project part is very in line with my PhD research topic, the activities I carried out are those described in Section 5.3.
  2) Developing an approach, illustrated in Figure A.1, for the system's safety assessment, i.e., the procedure that identifies and scores the potential risks associated to each component of the system. Due to the pres-

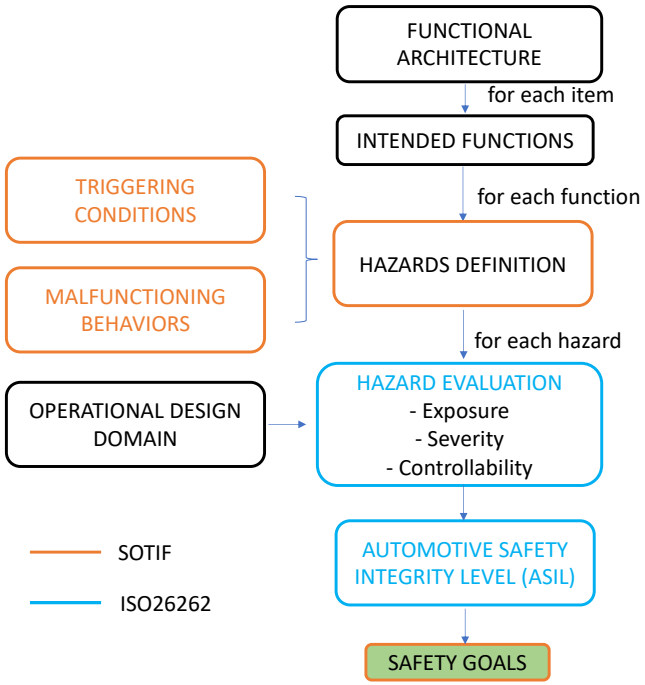
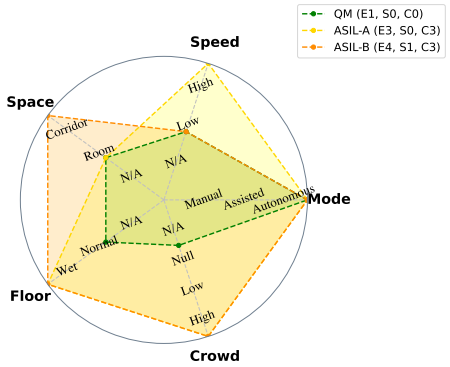


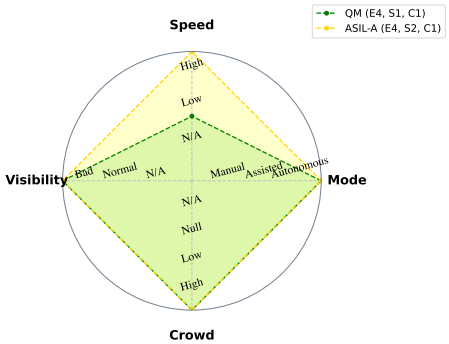
Figure A.1: High-level overview of the safety analysis approach within REXASI-PRO framework.

ence of AI elements, traditional approaches and standards like FMEA, IEC61508 [64] and ISO26262 [68] have been coupled together with SOTIF [70]. The overall outlined process involved hazards identification, by analyzing the limitations and/or conditions potentially leading to hazards (i.e., *malfunctioning behaviors*) and evaluating their impact on the system - in terms of exposure, severity and controllability - under environmental or wheelchair-related conditions as determined by the relevant ODD for each component. The last step was the definition of *safety goals* i.e., the strategies adopted to address and mitigate the potential risks identified during the analysis to ensure safe operations within the ODD.



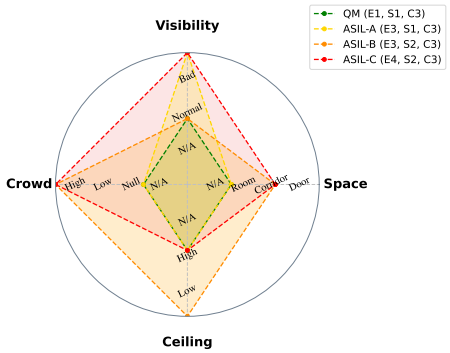
Malfunctioning behavior: the video monitoring system (VMS) fails to warn the wheelchair about an approaching person.
Hazard: the wheelchair does not brake when it should.
Potential effect: collision of the wheelchair with obstacles and/or people; injuries of the user.
Safety goal: the detection performed in the VMS will be based on several frames to give more precise and trustworthy information to the wheelchair.

(a) Cameras



Malfunctioning behavior: a sensor on the wheelchair does not perceive the environment correctly and fails in detecting a person.
Hazard: the wheelchair collides with people.
Potential effect: the user and/or other people can get injured because of the collision.
Safety goal: the wheelchair will be equipped with: a watchdog checking availability and plausibility of sensor data; a dead man switch that results in an immediate full stop in automatic driving.

(b) Wheelchair



Malfunctioning behavior: not detecting or wrongly detecting an obstacle.
Hazard: the flying robot crashes against obstacles.
Potential effect: potential damages to wheelchair user or to the battery of the drone.
Safety goal: the flying robot will have redundant sensors or will be featured with an additional protective case around the propeller; also, a safety driver will take control in emergency cases.

(c) Flying robot

Figure A.2: Examples from REXASI-PRO safety analysis. Each spider chart shows how different combinations of values for the ODD variables (reported in bold) determine a different ASIL level of risk for the individuated malfunctioning behaviors and hazards (QM: quality management; ASIL-A: low risk; ASIL-B: moderate risk; ASIL-C: high risk). ASIL is in turn assigned from exposure, severity and controllability scores. Risk levels other than QM require the introduction of safety goals to mitigate them.

Figure A.2 illustrates some sample outcomes of this process, for the three main technological components, i.e., the cameras (Fig. A.2a), the wheelchair (Fig. A.2b), and the flying robot (Fig. A.2c). For each of them, malfunctions that may lead to potentially dangerous hazards

have been identified. According to the conditions resulting from each particular combination of the variables in the respective ODD, the following scores have been assigned to each hazard: i) *exposure*, ranging from E0 (incredibly unlikely) to E4 (high probability), associated to the probability of encountering the hazard in a particular scenario (e.g., the probability of missed detections by cameras is larger when the floor is wet and the space is crowded than when being on a normal floor and with no people around); ii) *severity*, spanning from S0 (no injuries) to S4 (life-threatening/fatal injuries), is associated to the entity of the impact that the hazards might have on people (e.g., for the same visibility, crowd and mode, the hazard may have a worse impact when the wheelchair moves faster, as highlighted in A.2b); iii) *controllability*, from generally controllable (C0) to difficult to control/uncontrollable (C3), measuring the degree under which human intervention can take the control over the hazardous event (e.g., when moving at a high speed or in a tight space, it can be more difficult to control the movement). Finally, the combination of the assigned levels to the three metrics retrieves the corresponding ASIL. Whenever a risk level is different from QM, safety requirements are identified by defining suitable safety goals.

These are just examples and, although from the system-level reliability perspective, a clear connection with model-level reliability can be found in the strategies adopted to solve the individuated safety goals. As a matter of fact, countermeasures to a given risk may be taken not only at a hardware level, e.g., extending the sensors (such as the addition of sensors to the wheelchair or the flying robot), but also at the software – or, more precisely, at the AI software – level (e.g.,through multi-frame decision-making as far as the camera-based detection is concerned). Also, strategies this may include restricting the working conditions (e.g., limiting the wheelchair speed) and/or extending the functional architecture with the safety regions [25].

A.2 Secondary contributions in Reliable AI

This appendix Section summarizes the contributions I carried out, in parallel with my main research line, to other topics of interest in Reliable AI.

A.2.1 Rule-based Out of Distribution detection

Out-of-distribution (OoD) detection is one of the fundamental problems in RAI, which consists in recognizing when a machine learning model operates in an environment that significantly deviates from the one seen during training. This can

occur since models are generally deployed in an open-world scenario [157].
Detecting OoD is crucial for Reliable AI because the performance of a model on OoD data might deteriorate compared to that of the training stage, and potentially lead to fatal consequences. When an OoD is acknowledged, the autonomous system should therefore envision either a reject option (i.e., not providing any decision for that sample) and handle over the control to humans or move to any other kind of failsafe fallback.
Several literature solutions proposed to address the OoD challenge are based on robust distributional assumptions on the feature space [79], or the knowledge of probability density functions (PDF) for incoming and outgoing data in the training phase [18]. Also, techniques to tackle the OoD problem in neural networks have been developed, including the OpenMax score [17], ODIN score [81], energy score [83] and others. All these methods, however, are often impractical or require heavy parameters settings in real-world scenarios.

In this context, rule-based classifiers can drive the design of a new method to perform the OoD detection, without any distributional assumption nor particular parameter setting.

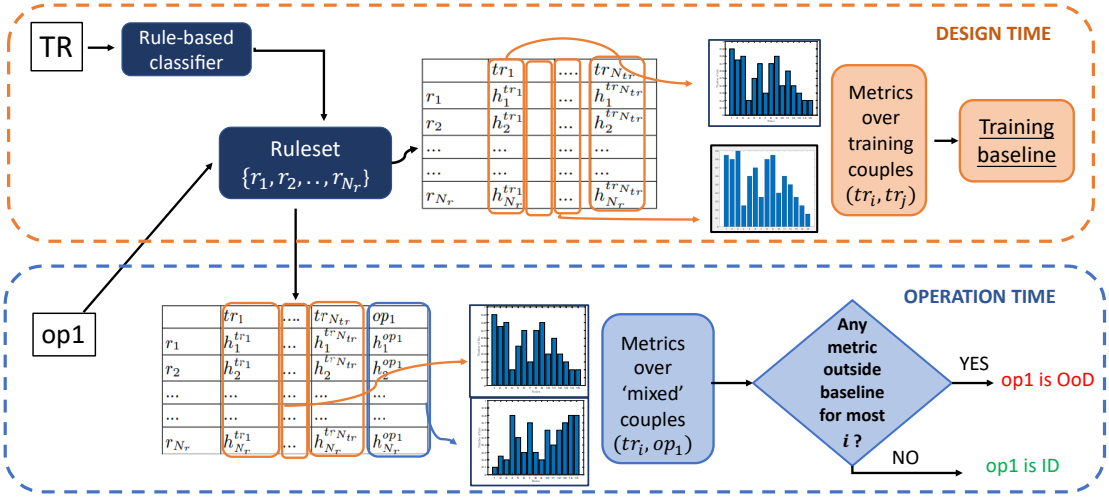


Figure A.3: Schematic summary of the rule-based OoD detection approach.

Consider a training dataset $TR = \{tr_1, \ldots, tr_{N_{tr}}\}$ divided into splits of size n_s, and a rule-based classifier $\mathcal{R}_{tr}$ with N_r rules trained on it; consider also an operational, unseen, set of data $op1$. The key concept behind the idea of exploiting rules for OoD purposes consists in computing the *number of rule hits* (or rule hits, in short), that is the frequency of validation of each rule by the split's samples. The collection

of rules hits over a single split determines a histogram:

$$\mathbf{h}^{(\cdot)} = \left\{h_i^{(\cdot)}\right\}, \quad h_i^{(\cdot)} \in [0,1], \; i = 1, \ldots, N_r,$$

where $(\cdot)$ denotes the considered split. Arguing that rule hits can be subject to changes while moving from data belonging to the training distribution to other that do not, the algorithm for OoD detection builds around the idea that quantitatively capturing such modifications in the rule hits, through computing some suitable metrics, it is possible to drive the detection of the OoD. Two phases are then foreseen, as also illustrated in Fig. A.3:

1. **Design time**: a *training baseline* is computed on the basis of the metrics computed over rule hits for training data only.

2. **Operation time**: the same metrics are computed over rule hits from both training and operational data, and the obtained values are compared to the baseline, finally providing a decision on the nature of the operational split, either in distribution (ID) or Out Of Distribution (OoD).

The adopted metrics are the ℓ_1 and ℓ_2 vector norms, as well as a modified version of mutual information (μI), the weighted mutual information $W\mu I$, which accounts for rule ordering within the histograms. The method was tested in three scenarios of interest (vehicle platooning, DNS and RUL estimation), with promising results in terms of FNR, which is the rate of missed OoD detections, as outlined in Table A.1.

Table A.1: Results on rule hits-based OoD detection according to l_1, l_2, μI and op_1 metrics.

		μI		$W\mu I$		l_1		l_2	
		Range	FNR [%]	Range	FNR [%]	Range	FNR [%]	Range	FNR [%]
Vehicle Platooning	$tr_{LOW} - tr_{LOW}$	[0.87, 2.73]		[0.02, 0.06]		[0.02, 0.12]		[0.01, 0.04]	
	$tr_{LOW} - op_{HIGH}$	[0.04, 0.97]	8	[0.51, 0.73]	0		0		0
DNS	$tr_{p2p} - tr_{p2p}$	[1.5, 2.2]		[0.01, 0.15]		[0.002, 0.090]		[0.002, 0.047]	
	$tr_{p2p} - op_{ssh}$	[0, 0.7]	0	[0.73, 0.96]	0	[1.630, 1.770]	0	[0.820, 0.890]	0
RUL	$tr_1 - tr_1$	[0.707, 1.442]		[0.023, 0.045]		[0.09, 0.20]		[0.02, 0.05]	
	$tr_1 - op_a$	[0.019, 0.027]	0	[0.291, 0.297]	0	[3.16, 3.26]	0	[1.05, 1.06]	0
	$tr_1 - op_b$	[0.182, 0.278]	0	[0.159, 0.179]	0	[1.16, 1.40]	0	[0.30, 0.34]	0
	$tr_1 - op_c$	[0.019, 0.027]	0	[0.290, 0.297]	0	[3.16, 3.26]	0	[1.05, 1.06]	0

My contribution in this work mostly focused on the preliminary assessment about if rule hits variations could actually be associated to data distribution changes, while the design of the overall algorithm and the metrics therein was not my primary contribution. Therefore, more formal methodological and experimental details are here omitted and I remind the interested reader to the related publications (Appendix B, [J7],[C5]).

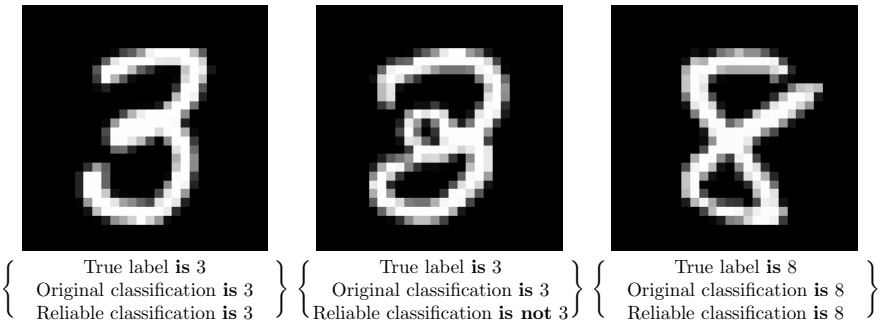


Figure A.4: Visual intuition behind the need for deep probabilistic scaling approach. The goal is to understand which samples are "not conformal" with respect to a target class.

A.2.2 Deep probabilistic scaling

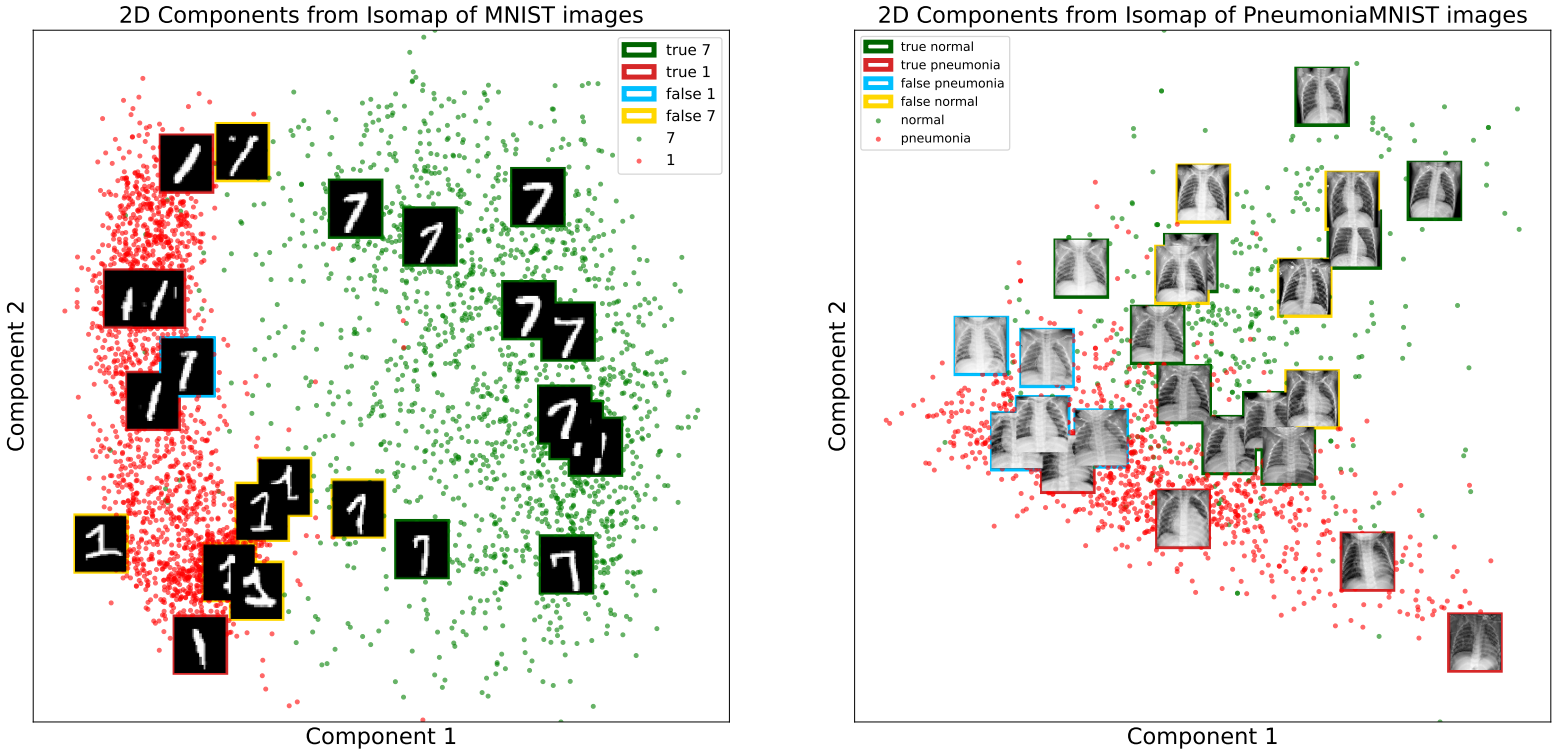
As introduced in Section 1.2.2, uncertainty quantification plays an important role in machine/deep learning [50, 2] and is an essential ingredient in reliable AI. Nowadays, binary image classification is one of the main tasks in AI-based computer vision, often through very complex model architectures and/or very large training datasets. For this reason, reliability assessment needs to develop innovative UQ tecniques that avoid computationally expensive retraining of such models.
In this context, a probabilistic method, called *deep probabilistic scaling (deep PS)*, based on the concept of adjustable classifiers (see also Eq. 5.3 from Sec. 5.3.3), is defined, allowing to bound the number of false negatives (or false positives) *without changing* the architecture of the network.
Consider the decision function $\hat{f}$ of a Convolutional Neural Network (CNN):

$$\begin{aligned}\hat{f}(\boldsymbol{x}) &= \sigma(\boldsymbol{w}_L g_{L-1}(\boldsymbol{w}_{L-2} g_{L-2}(\dots g_1(\boldsymbol{w}_0 \boldsymbol{x} + b_0)\dots) + b_{L-1}) + b_L) - \frac{1}{2} \\ &= \sigma(h(\boldsymbol{x})) - \frac{1}{2}\end{aligned}$$

where L is the depth of the network, $\boldsymbol{w}_\ell$ and $g_\ell(\cdot)$ are respectively weights and activation function of the layer $\ell \in [L]$, $h(\cdot)$ is the function representing the composition of all hidden layers and $\sigma(\cdot)$ is the sigmoid function.
Then, the adjustable classifier

$$f(\boldsymbol{x}, \rho) = \hat{f}(\boldsymbol{x}) + \rho, \quad \rho \in \mathbb{R},$$

can be defined and, by following the approach detailed in [27], the optimal tuning ρ of the network's decision boundary can be individuated.
The goal is to identify a region in the input space, where the prediction corresponds with a user-defined target class with probability at least $1 - \varepsilon$, where $\varepsilon \in (0,1)$ is the user-defined error level.



(*a*) MNIST1-7. (*b*) PneumoniaMNIST.

Figure A.5: Distribution of uncertainty of MNIST17 and PneumoniaMNIST. Shown in blue and yellow, respectively, are the uncertain classifications for class *'1'* and class *'7'* for MNIST1-7 and for class "*normal*" and class "*pneumonia*" for PneumoniaMNIST.

Figure A.4 provides an intuition on the practical implications of this method: the numbers 3 and 8 in the MNIST dataset [35] can be easily confused by the human eye, and this is understandable because the digits appear pretty similar to each other. Deep SC then translates this idea of similarity into an "intrinsic probability" that the samples have: the digit on the left part of the image has a high probability of being a 3 whereas the middle digit, while labeled as a 3, has clearly more features in common with an 8 and thus a lower probability of being identified as a 3.
The experimental tests were based on meaningful binary problems identified in benchmark datasets, that is MNIST: '1' vs '7', '2' vs '5' and '3' vs '8'; CIFAR10 [76]: truck vs automobile; PneumoniaMNIST[156]: normal vs pneumonia. In all cases, the classification error of a pretrained CNN model improved after Deep SC

and was bounded by the ε value that was set.
Figure A.5, related to these tests, visually confirms the intuitions of Figure A.4.
In this work, my personal contribution was in support to the experimental evaluation phase: from the choice of the datasets and creation of the binary classification problems, to training of the base image classification models, and visualization of the results. For this reason, methodological details are here omitted, referring the reader to the related publication [C8] from Appendix B.

A.3 Research period at DLR

This Section briefly outlines the activity I carried out during my three months spent at the German Aerospace Center (DLR), Institute of Flight Systems, Braunschweig (Germany), in the third PhD year.

Certification of Artificial Intelligence in the avionics field poses many challenges, especially when humans are involved in Unmanned Aerial Vehicles (UAVs) flight operations, e.g., in emergency medicine scenarios, search and rescue or dropping goods, since failures of AI-guided detection systems might result in severe harms to people. Despite often revealing highly accurate, deep learning-based people detection models have a black-box nature, preventing the possibility of understanding why the model generated its outcomes and, subsequently, analyzing the reasons for correct results and failures.
In this context, and in compliance with Trustworthy Artificial Intelligence principle of transparency, XAI - and, in particular, rule-based models - can have an important role. The work I carried out at DLR attempts to address these issues, by investigating the innovative use of rule-based classifiers as a transparent validation tool of a deep learning model for people detection in aerial images. More specifically, the objective is to obtain a set of interpretable if-then rules characterizing the space of image features associated with a good or bad performance of that model. Besides shedding light on the logic of the humans detection successes and failures, these rules can serve as a performance monitor at runtime, by triggering alerts in case input images do not satisfy them.
The overall idea of the work is shown in Figure A.6
Based on features extracted from aerial images of people, I mostly focused on the following phases (conceptualization and testing):

- Feature selection via Pearson's correlation analysis [126] for the choice of the most suitable set of features to train an effective and stable rule-based XAI classifier;
- Rule-based classifier training: generation of a suitable set of if-then rules; different models were tried: Logic Learning Machine (2.2.3), decision tree (2.2.4), skope-rules (2.2.5);

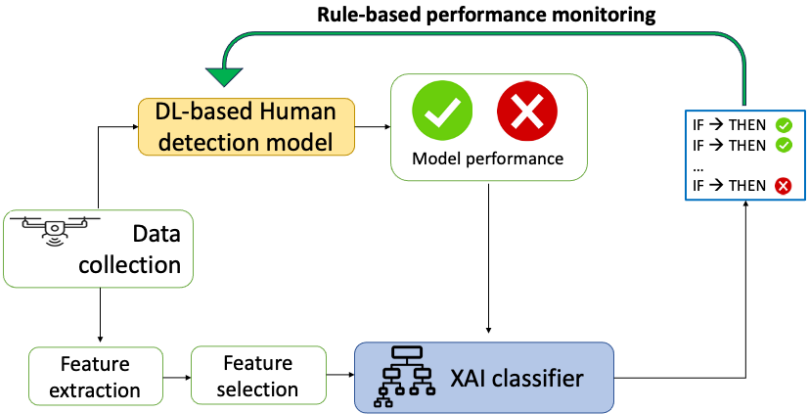


Figure A.6: Flowchart of the work. A deep learning-based humans detection model is applied to images collected from a UAV. Features extracted from the images, combined with information on the good or bad detection performance, are fed to a XAI classifier, generating a set of rules for monitoring

- Rules validation via a χ^2 test to assess their statistical dependence with the output [151].

The obtained results, in terms of rule generation, are still preliminary as far as it regards reliability: rules themselves arise from a ML model that, even though explainable, is subject to uncertainty and needs to be handled before being able to use the rules in practice. So, how to avoid that the error of rule generation further propagates and reflects in the human detection performance monitoring? Moreover, DNN-based object detection tasks often involve long training processes by using images from different sources, either real or even synthetic. And rule generation from real versus synthetic data is a matter of discussion around the kind of knowledge one can derive from rules, e.g., if rules on real versus synthetic data do not match, would the latter have to be considered 'wrong'? Or would it mean that new plausible factors are being discovered?
Further work is then needed to address these issues.

Appendix B

Publications

This Appendix reports the publications I achieved during the three PhD years.

Journals

[J1] **S. Narteni**, V. Orani, E. Cambiaso, M. Rucco and M. Mongelli, "On the Intersection of Explainable and Reliable AI for Physical Fatigue Prediction," in IEEE Access, vol.10, 76243-76260, 2022, doi: 10.1109/ACCESS.2022.3191907.

[J2] **S. Narteni**, V. Orani, I. Vaccari, E. Cambiaso and M. Mongelli, "Sensitivity of Logic Learning Machine for Reliability in Safety-Critical Systems," in IEEE Intelligent Systems, vol.37, 66-74, 2022, doi:10.1109/MIS.2022.3159098.

[J3] 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, 83949-83970, 2022, doi: 10.1109/ACCESS.2022.3197299.

[J4] M. Lenatti, **S. Narteni**, A. Paglialonga, V. Rampa, M. Mongelli, "Dual-View Single-Shot Multibox Detector at Urban Intersections: Settings and Performance Evaluation". Sensors. 2023; 23(6):3195. doi: 10.3390/s23063195.

[J5] **S. Narteni**, M. Muselli, F. Dabbene, M. Mongelli "Trustworthy artificial intelligence classification-based equivalent bandwidth control", Computer Communications, vol. 209, pp. 260-272, 2023. doi: 10.1016/j.comcom.2023.07.005.

[J6] **S. Narteni**, I. Baiardini, F. Braido, M. Mongelli "Explainable artificial intelligence for cough-related quality of life impairment prediction in asthmatic patients". PLOS ONE 19(3): e0292980, 2024, doi: 10.1371/journal.pone.0292980.

[J7] G. De Bernardi, **S. Narteni**, E. Cambiaso and M. Mongelli, "Rule-Based Out-of-Distribution Detection," in IEEE Transactions on Artificial Intelligence, vol. 5, no. 6, pp. 2627-2637, 2024, doi: 10.1109/TAI.2023.3323923.

[J8] E. Cambiaso, **S. Narteni**, I. Baiardini, F. Braido, A. Paglialonga, M. Mongelli. "Advancements on IoT and AI applied to Pneumology". Microprocessors and Microsystems, 108, 105062, 2024, doi: 10.1016/j.micpro.2024.105062.

[J9] **S. Narteni**, V. Orani, E. Ferrari, D. Verda, E. Cambiaso, M. Mongelli, "Explainable Evaluation of Generative Adversarial Networks for Wearables Data Augmentation", Engineering Applications of Artificial Intelligence,145, 110133, 2025, doi: 10.1016/j.engappai.2025.110133.

[J10] **S. Narteni**, A. Carlevaro, F. Dabbene, M. Muselli, M. Mongelli. "A novel score function for conformal prediction in rule-based binary classification". Pattern Recognition, 2024 [*under review*].

[J11] **S. Narteni**, A. Carlevaro, J. Guzzi, M. Mongelli "Explainable data-driven confidence regions for safe and efficient robotic navigation" Reliability Engineering and Systems Safety, 2024 [*submitted*]

Conferences

[C1] **S. Narteni**, M. Ferretti, V. Orani, I. Vaccari, E. Cambiaso, M. Mongelli "From Explainable to Reliable Artificial Intelligence". In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds) Machine Learning and Knowledge Extraction. CD-MAKE 2021, 2021. Lecture Notes in Computer Science(), vol 12844. Springer, Cham. doi: 10.1007/978-3-030-84060-0_17.

[C2] 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), New York, NY, USA, 2022, pp. 1-6, doi: 10.1109/INFOCOMWKSHPS54753.2022.9797955.

[C3] **S. Narteni**, M. Ferretti, V. Rampa, M. Mongelli. "Bag-of-Words Similarity in eXplainable AI". In: Arai, K. (eds) Intelligent Systems and Applications. IntelliSys 2022. Lecture Notes in Networks and Systems, vol 543. Springer, Cham. doi: 10.1007/978-3-031-16078-3_58

[C4] **S. Narteni**, V. Orani, E. Ferrari, D. Verda, E. Cambiaso and M. Mongelli, "A New XAI-based Evaluation of Generative Adversarial Networks for IMU Data Augmentation," 2022 IEEE International Conference on E-health Networking, Application & Services (HealthCom), Genoa, Italy, 2022, pp. 167-172, doi: 10.1109/HealthCom54947.2022.9982780.

[C5] G. De Bernardi, **S. Narteni**, E. Cambiaso, M. Muselli, M. Mongelli. "Weighted Mutual Information for Out-Of-Distribution Detection". In: Longo, L. (eds) Explainable Artificial Intelligence. xAI 2023, 2023. Communications in Computer and Information Science, vol 1903. Springer, Cham. doi: 10.1007/978-3-031-44070-0_16

[C6] **S. Narteni**, A. Carlevaro, F. Dabbene, M. Muselli, M. Mongelli. "CONFIDERAI: CONFormal Interpretable-by-Design score function for Explainable and Reliable Artificial Intelligence" Proceedings of the Twelfth Symposium on Conformal and Probabilistic Prediction with Applications, PMLR 204:485-487, 2023. https://proceedings.mlr.press/v204/narteni23a.html

[C7] **S. Narteni**, A. Carlevaro, J. Guzzi, M. Mongelli. "Ensuring Safe Social Navigation via Explainable Probabilistic and Conformal Safety Regions". In: Longo, L., Lapuschkin, S., Seifert, C. (eds) Explainable Artificial Intelligence. xAI 2024. Communications in Computer and Information Science, vol 2156. Springer, Cham. doi: 10.1007/978-3-031-63803-9_22

[C8] A. Carlevaro, **S. Narteni**, F. Dabbene, T. Alamo, M. Mongelli. "A probabilistic scaling approach to conformal predictions in binary image classification". Proceedings of the Thirteenth Symposium on Conformal and Probabilistic Prediction with Applications, PMLR 230:28-43, 2024. https://proceedings.mlr.press/v230/carlevaro24a.html

[C9] **S. Narteni**, M. Mongelli, J. Rüter, C. Torens, U. Durak. "Transparent Assessment of Automated Human Detection in Aerial Images via Explainable AI". AIAA Scitech Forum 2025. doi: 10.2514/6.2025-2676.

Nomenclature

Acronyms / Abbreviations

AI Artificial Intelligence

AML Adversarial Machine Learning

BoW Bag of Words

CCS Conformal Critical Set

CP Conformal Prediction

CSR Conformal Safety Region

CW Carlini-Wagner

DT Decision Tree

FGSM Fast Gradient Sign Method

FNR False Negative Rate

FN False Negatives

FPR False Positive Rate

JSMA Jacobian-based Saliency Map

LLM Logic Learning Machine

ML Machine Learning

PSR Probabilistic Safety Region

RAI Reliable Artificial Intelligence

TAI Trustworthy Artificial Intelligence

TNR True Negative Rate

TPR True Positive Rate

UQ Uncertainty Quantification

XAI Explainable Artificial Intelligence

Bibliography

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

[2] Moloud Abdar et al. "A review of uncertainty quantification in deep learning: Techniques, applications and challenges". In: *Information fusion* 76 (2021), pp. 243–297. DOI: 10.1016/j.inffus.2021.05.008.

[3] Husam Abdelqader et al. "Interpretable and Reliable Rule Classification Based on Conformal Prediction". In: *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part I*. Springer. 2023, pp. 385–401. DOI: 10.1007/978-3-031-23618-1_26.

[4] Amina Adadi and Mohammed Berrada. "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6 (2018), pp. 52138–52160. DOI: 10.1109/ACCESS.2018.2870052.

[5] 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. DOI: 10.1002/dac.2836.

[6] Naveed Akhtar et al. "Attack to Fool and Explain Deep Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), pp. 1–1. DOI: 10.1109/TPAMI.2021.3083769.

[7] Sajid Ali et al. "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence". In: *Information fusion* 99 (2023), p. 101805. DOI: 10.1016/j.inffus.2023.101805.

[8] Turki Alsuwian, Rana Basharat Saeed, and Arslan Ahmed Amin. "Autonomous Vehicle with Emergency Braking Algorithm Based on Multi-Sensor Fusion and Super Twisting Speed Controller". In: *Applied Sciences* 12.17 (2022). ISSN: 2076-3417. DOI: 10.3390/app12178458.

[9] M. Ammour, R. Orjuela, and M. Basset. "Collision avoidance for autonomous vehicle using MPC and time varying Sigmoid safety constraints∗∗This work is supported by the Excellence Program of the Algerian Government." In: *IFAC-PapersOnLine* 54.10 (2021). 6th IFAC Conference on Engine Powertrain Control, Simulation and Modeling E-COSM 2021, pp. 39–44. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2021.10.138.

[10] Elaine Angelino et al. "Learning Certifiably Optimal Rule Lists". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. Halifax, NS, Canada: Association for Computing Machinery, 2017, pp. 35–44. ISBN: 9781450348874. DOI: 10.1145/3097983.3098047. URL: https://doi.org/10.1145/3097983.3098047.

[11] Anastasios N. Angelopoulos and Stephen Bates. *A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification*. 2021. DOI: 10.48550/ARXIV.2107.07511. URL: https://arxiv.org/abs/2107.07511.

[12] Anastasios N. Angelopoulos and Stephen Bates. "Conformal Prediction: A Gentle Introduction". In: *Foundations and Trends® in Machine Learning* 16.4 (2023), pp. 494–591. ISSN: 1935-8237. DOI: 10.1561/2200000101.

[13] https://www.asam.net/index.php?eID=dumpFile&t=f&f=4303&token=3135965e578e5bb92a01725cd37823c3979da158.

[14] I Baiardini et al. "A new tool to assess and monitor the burden of chronic cough on quality of life: Chronic Cough Impact Questionnaire". In: *Allergy* 60.4 (2005), pp. 482–488. DOI: 10.1111/j.1398-9995.2005.00743.x.

[15] Sarah Adel Bargal et al. "Guided Zoom: Zooming into Network Evidence to Refine Fine-Grained Model Decisions". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (2021), pp. 4196–4202. DOI: 10.1109/TPAMI.2021.3054303.

[16] 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.

[17] Abhijit Bendale and Terrance Boult. "Towards Open World Recognition". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1893–1902. DOI: 10.1109/CVPR.2015.7298799.

[18] Julian Bitterwolf et al. "Breaking down out-of-distribution detection: Many methods based on ood training data estimate a combination of the same core quantities". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 2041–2074. URL: https://proceedings.mlr.press/v162/bitterwolf22a/bitterwolf22a.pdf.

[19] *Body Signals of Smoking*. URL: https://www.kaggle.com/datasets/kukuroo3/body-signal-of-smoking?select=smoking.csv.

[20] Zoran Bosnić and Igor Kononenko. "An overview of advances in reliability estimation of individual predictions in machine learning". In: *Intelligent Data Analysis* 13.2 (2009), pp. 385–401. DOI: 10.3233/IDA-2009-0371.

[21] Lukas Brunke et al. "Safe learning in robotics: From learning-based control to safe reinforcement learning". In: *Annual Review of Control, Robotics, and Autonomous Systems* 5 (2022), pp. 411–444. DOI: 10.1146/annurev-control-042920-020211.

[22] *Cardiovascular disease dataset*. URL: https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset.

[23] Alberto Carlevaro and Maurizio Mongelli. "A New SVDD Approach to Reliable and Explainable AI". In: *IEEE Intelligent Systems* 37.2 (2022), pp. 55–68. DOI: 10.1109/MIS.2021.3123669.

[24] Alberto Carlevaro et al. "A probabilistic scaling approach to conformal predictions in binary image classification". In: *The 13th Symposium on Conformal and Probabilistic Prediction with Applications*. PMLR. 2024, pp. 28–43.

[25] Alberto Carlevaro et al. "ARE DIGITAL TWINS SUITABLE TO DRIVE SAFE AI?" In: *BUILD-IT 2023 Workshop*. Rome, Italy, 19-20 October 2023.

[26] Alberto Carlevaro et al. "Conformal predictions for probabilistically robust scalable machine learning classification". In: *Machine Learning* 113 (2024), pp. 6645–6661. DOI: 10.1007/s10994-024-06571-6.

[27] Alberto Carlevaro et al. "Probabilistic Safety Regions Via Finite Families of Scalable Classifiers". In: *arXiv preprint arXiv:2309.04627* (2023). DOI: 10.48550/arXiv.2309.04627.

[28] 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. DOI: 10.1109/SP.2017.49.

[29] Fernando Castaneda Garcia-Rozas. "Safe Control of Robotic Systems under Uncertainty: Reconciling Model-based and Data-driven Methods". PhD thesis. UC Berkeley, 2023. URL: https://hybrid-robotics.berkeley.edu/publications/Dissertation2023_Fernando_Castaneda.pdf.

[30] Vinay Chamola et al. “A Review of Trustworthy and Explainable Artificial Intelligence (XAI)”. In: *IEEE Access* 11 (2023), pp. 78994–79015. DOI: 10.1109/ACCESS.2023.3294569.

[31] Peter Clark and Tim Niblett. “The CN2 induction algorithm”. In: *Machine learning* 3.4 (1989), pp. 261–283. DOI: 10.1007/BF00116835.

[32] William W Cohen. “Learning to classify English text with ILP methods”. In: *Advances in inductive logic programming* 32 (1995), pp. 124–143.

[33] Anthony Corso et al. “A holistic assessment of the reliability of machine learning systems”. In: *arXiv preprint arXiv:2307.10586* (2023). DOI: 10.48550/arXiv.2307.10586.

[34] David Dalrymple et al. “Towards Guaranteed Safe AI: A Framework for Ensuring Robust and Reliable AI Systems”. In: *arXiv preprint arXiv:2405.06624* (2024). DOI: 10.48550/arXiv.2405.06624.

[35] Li Deng. “The mnist database of handwritten digit images for machine learning research”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[36] Natalia Díaz-Rodríguez et al. “Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation”. In: *Information Fusion* 99 (2023), p. 101896. DOI: 10.1016/j.inffus.2023.101896.

[37] Tuan Q. Dinh et al. “Performing Group Difference Testing on Graph Structured Data From GANs: Analysis and Applications in Neuroimaging”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.2 (2022), pp. 877–889. DOI: 10.1109/TPAMI.2020.3013433.

[38] Rudresh Dwivedi et al. “Explainable AI (XAI): Core ideas, techniques, and solutions”. In: *ACM Computing Surveys* 55.9 (2023), pp. 1–33. DOI: 10.1145/3561048.

[39] *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.

[40] *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.

[41] *EEG Eye State*. URL: https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State.

[42] European Commission. *Proposal for a Regulation Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act)*. European Commission. Accessed: 2024-10-22. 2021. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206.

[43] Alberto Fernández et al. "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary". In: *Journal of artificial intelligence research* 61 (2018), pp. 863–905. DOI: 10.1613/jair.1.11192.

[44] 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. DOI: 10.1007/978-3-031-16281-7_4.

[45] Jerome H. Friedman and Bogdan E. Popescu. "Predictive learning via rule ensembles". In: *The Annals of Applied Statistics* 2.3 (2008), pp. 916–954. DOI: 10.1214/07-AOAS148.

[46] Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of rule learning*. Springer Berlin, Heidelberg, 2012. DOI: 10.1007/978-3-540-75197-7.

[47] M. Galar et al. "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.4 (2012), pp. 463–484. DOI: 10.1109/TSMCC.2011.2161285.

[48] Alfredo García, David Llopis-Castelló, and Francisco Javier Camacho-Torregrosa. "From the vehicle-based concept of operational design domain to the road-based concept of operational road section". In: *Frontiers in Built Environment* 8 (2022), p. 901840. DOI: 10.3389/fbuil.2022.901840.

[49] Arturo Garcia-Garcia, Marek Z. Reformat, and Andres Mendez-Vazquez. "Similarity-based method for reduction of fuzzy rules". In: *2016 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS)*. 2016, pp. 1–6. DOI: 10.1109/NAFIPS.2016.7851603.

[50] Jakob Gawlikowski et al. "A survey of uncertainty in deep neural networks". In: *Artificial Intelligence Review* 56.Suppl 1 (2023), pp. 1513–1589.

[51] Ian Goodfellow et al. "Generative adversarial networks". In: *Commun. ACM* 63.11 (Oct. 2020), pp. 139–144. ISSN: 0001-0782. DOI: 10.1145/3422622.

[52] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014). DOI: 10.48550/arXiv.1412.6572.

[53] Mara Graziani et al. “A global taxonomy of interpretable AI: unifying the terminology for the technical and social sciences”. In: *Artificial intelligence review* 56.4 (2023), pp. 3473–3504. DOI: 10.1007/s10462-022-10256-8.

[54] Riccardo Guidotti et al. “A survey of methods for explaining black box models”. In: *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42. DOI: https://doi.org/10.1145/3236009.

[55] Zhen Guo et al. “A survey on uncertainty reasoning and quantification in belief theory and its application to deep learning”. In: *Information Fusion* 101 (2024), p. 101987. DOI: 10.1016/j.inffus.2023.101987.

[56] Jérôme Guzzi et al. “Human-friendly robot navigation in dynamic environments”. In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 423–430. DOI: 10.1109/ICRA.2013.6630610.

[57] Adib Habbal, Mohamed Khalif Ali, and Mustafa Ali Abuzaraida. “Artificial Intelligence Trust, risk and security management (AI trism): Frameworks, applications, challenges and future research directions”. In: *Expert Systems with Applications* 240 (2024), p. 122442. DOI: 10.1016/j.eswa.2023.122442.

[58] Richard Hawkins et al. “Guidance on the assurance of machine learning in autonomous systems (AMLAS)”. In: *arXiv preprint arXiv:2102.01564* (2021). DOI: 10.48550/arXiv.2102.01564.

[59] Martin Heusel et al. “Gans trained by a two time-scale update rule converge to a local nash equilibrium”. In: *Advances in neural information processing systems* 30 (2017).

[60] High-Level Expert Group on AI. *Ethics guidelines for trustworthy AI*. eng. Report. Brussels: European Commission, Apr. 2019. URL: https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai.

[61] Shoji Hirano and Shusaku Tsumoto. “Detection of differences between syntactic and semantic similarities”. In: *Rough Sets and Current Trends in Computing: 4th International Conference, RSCTC 2004, Uppsala, Sweden, June 1-5, 2004. Proceedings 4*. Springer. 2004, pp. 529–538. DOI: 10.1007/978-3-540-25929-9_64.

[62] Zhonglin Hou et al. “Assessing Unknown Hazards for SOTIF Based on Twin Scenarios Empowered Autonomous Driving”. In: *IEEE Internet of Things Journal* 11.20 (2024), pp. 32631–32644. DOI: 10.1109/JIOT.2024.3424550.

[63] Eyke Hüllermeier and Willem Waegeman. “Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods”. In: *Machine learning* 110.3 (2021), pp. 457–506. DOI: 10.1007/s10994-021-05946-3.

[64] https://webstore.iec.ch/en/publication/6007.

[65] "IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries". In: *IEEE Std 610* (1991), pp. 1–217. DOI: 10.1109/IEEESTD.1991.106963.

[66] https://2018.ds3-datascience-polytechnique.fr/wp-content/uploads/2018/06/DS3-309.pdf.

[67] https://www.iso.org/standard/72704.html.

[68] https://www.iso.org/standard/68383.html.

[69] *Standardization in the area of Artificial Intelligence*. Standard. https://www.iso.org/committee/6794475.html. Washington, DC 20036, USA, Creation date 2017.

[70] https://www.iso.org/standard/77490.html.

[71] Yan Jia et al. "The Role of Explainability in Assuring Safety of Machine Learning in Healthcare". In: *IEEE Transactions on Emerging Topics in Computing* 10.4 (2022), pp. 1746–1760. DOI: 10.1109/TETC.2022.3171314.

[72] Bhavya Kailkhura et al. "Reliable and explainable machine-learning methods for accelerated material discovery". In: *npj Computational Materials* 5.1 (2019), p. 108. DOI: 10.1038/s41524-019-0248-2.

[73] Eoin M Kenny et al. "Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies". In: *Artificial Intelligence* 294 (2021), p. 103459. DOI: 10.1016/j.artint.2021.103471.

[74] Noam Kolt. "Algorithmic black swans". In: *Washington University Law Review* 101 (2023).

[75] T. Kontogiannis, M.C. Leva, and N. Balfe. "Total Safety Management: Principles, processes and methods". In: *Safety Science* 100 (2017). SAFETY: Methods and applications for Total Safety Management, pp. 128–142. ISSN: 0925-7535. DOI: https://doi.org/10.1016/j.ssci.2016.09.015.

[76] Alex Krizhevsky, Geoffrey Hinton, et al. *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto, ON, Canada, 2009.

[77] W. Kuo and M.J. Zuo. *Optimal Reliability Modeling: Principles and Applications*. Wiley, 2003. ISBN: 9780471275459.

[78] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. "Interpretable decision sets: A joint framework for description and prediction". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1675–1684. DOI: 10.1145/2939672.2939874.

[79] Kimin Lee et al. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks". In: *Advances in neural information processing systems* 31 (2018). URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/abdeb6f575ac5c6676b747bca8d09cc2-Paper.pdf.

[80] Benjamin Letham et al. "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model". In: *The Annals of Applied Statistics* 9.3 (2015), pp. 1350–1371.

[81] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. "Enhancing the reliability of out-of-distribution image detection in neural networks". In: *arXiv preprint arXiv:1706.02690* (2017).

[82] Tsung-Yi Lin et al. "Feature Pyramid Networks for Object Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 936–944. DOI: 10.1109/CVPR.2017.106.

[83] Weitang Liu et al. "Energy-based out-of-distribution detection". In: *Advances in neural information processing systems* 33 (2020), pp. 21464–21475.

[84] Luca Longo et al. "Explainable Artificial Intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions". In: *Information Fusion* 106 (2024), p. 102301. DOI: 10.1016/j.inffus.2024.102301.

[85] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[86] *Magic gamma telescope dataset*. URL: https://www.kaggle.com/datasets/abhinand05/magic-gamma-telescope-dataset.

[87] Martina Mammarella et al. "Chance-constrained sets approximation: A probabilistic scaling approach". In: *Automatica* 137 (2022), p. 110108. DOI: 10.1016/j.automatica.2021.110108.

[88] Christian Mandel, Tim Laue, and Serge Autexier. "Chapter 12 - Smart-wheelchairs". In: *Smart Wheelchairs and Brain-Computer Interfaces*. Ed. by Pablo Diez. Academic Press, 2018, pp. 291–322. ISBN: 978-0-12-812892-3. DOI: 10.1016/B978-0-12-812892-3.00012-1.

[89] Alan D Mead. "Psychometric reliability: Definition, estimation, and application". In: *Wiley StatsRef: Statistics Reference Online* (2014), pp. 1–6. DOI: 10.1002/9781118445112.stat06409.pub2.

[90] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: *CoRR* abs/1411.1784 (2014). DOI: 10.48550/arXiv.1411.1784. arXiv: 1411.1784. URL: http://arxiv.org/abs/1411.1784.

[91] Sina Mohseni et al. "Practical Solutions for Machine Learning Safety in Autonomous Vehicles". In: *SafeAI@AAAI*. 2020.

[92] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. 2022. URL: https://christophm.github.io/interpretable-ml-book.

[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). DOI: 10.1049/iet-cps.2018.5055.

[94] 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. DOI: 10.1049/iet-cps.2018.5055.

[95] Mehdi Moussaid, Dirk Helbing, and Guy Theraulaz. "How simple rules determine pedestrian behavior and crowd disasters". In: *Proceedings of the National Academy of Sciences* 108.17 (2011), pp. 6884–6888. DOI: 10.1073/pnas.1016507108.

[96] M. Muselli and A. Quarati. "Reconstructing positive Boolean functions with shadow clustering". In: *Proceedings of the 2005 European Conference on Circuit Theory and Design, 2005.* Vol. 3. 2005, III/377–III/380 vol. 3. DOI: 10.1109/ECCTD.2005.1523139.

[97] Marco Muselli. *Switching Neural Networks: A New Connectionist Model for Classification*. Jan. 2005. DOI: 10.1007/11731177_4.

[98] Marco Muselli and Enrico Ferrari. "Coupling Logical Analysis of Data and Shadow Clustering for Partially Defined Positive Boolean Function Reconstruction". In: *IEEE Transactions on Knowledge and Data Engineering* 23.1 (2011), pp. 37–50. DOI: 10.1109/TKDE.2009.206.

[99] Sara Narteni et al. "Bag-of-Words Similarity in eXplainable AI". In: *Proceedings of SAI Intelligent Systems Conference*. Springer. 2022, pp. 835–851. DOI: 10.1007/978-3-031-16078-3_58.

[100] Robert A Nathan et al. "Development of the asthma control test: a survey for assessing asthma control". In: *Journal of Allergy and Clinical Immunology* 113.1 (2004), pp. 59–65. DOI: 10.1016/j.jaci.2003.09.008.

[101] Venkat Nemani et al. "Uncertainty quantification in machine learning for engineering design and health prognostics: A tutorial". In: *Mechanical Systems and Signal Processing* 205 (2023), p. 110796. DOI: 10.1016/j.ymssp.2023.110796.

[102] Giovanna Nicora et al. "Evaluating pointwise reliability of machine learning prediction". In: *Journal of Biomedical Informatics* 127 (2022), p. 103996. DOI: 10.1016/j.jbi.2022.103996.

[103] *PAMAP2 Physical Activity Monitoring Dataset*. https://archive.ics.uci.edu/ml/datasets/PAMAP2+Physical+Activity+Monitoring.

[104] Danqing Pan, Tong Wang, and Satoshi Hara. "Interpretable companions for black-box models". In: *International conference on artificial intelligence and statistics*. PMLR. 2020, pp. 2444–2454. URL: https://proceedings.mlr.press/v108/pan20a/pan20a.pdf.

[105] Cecilia Panigutti et al. "The role of explainable AI in the context of the AI Act". In: *Proceedings of the 2023 ACM conference on fairness, accountability, and transparency*. 2023, pp. 1139–1150. DOI: 10.1145/3593013.3594069.

[106] 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. DOI: 10.1109/EuroSP.2016.36.

[107] Stefano Parodi et al. "Differential diagnosis of pleural mesothelioma using Logic Learning Machine". In: *BMC bioinformatics* 16 Suppl 9 (June 2015), S3. DOI: 10.1186/1471-2105-16-S9-S3.

[108] Stefano Parodi et al. "Identifying Environmental and Social Factors Predisposing to Pathological Gambling Combining Standard Logistic Regression and Logic Learning Machine". In: *Journal of Gambling Studies* 33.4 (2017), pp. 1121–1137. DOI: 10.1007/s10899-017-9679-1.

[109] Stefano Parodi et al. "Logic Learning Machine and standard supervised methods for Hodgkins lymphoma prognosis using gene expression data and clinical variables". In: *Health Informatics Journal* 24 (June 2016). DOI: 10.1177/1460458216655188.

[110] Abdul Wahab Qurashi, Violeta Holmes, and Anju P. Johnson. "Document Processing: Methods for Semantic Text Similarity Analysis". In: *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. 2020, pp. 1–6. DOI: 10.1109/INISTA49547.2020.9194665.

[111] M. Mostafizur Rahman and D. N. Davis. "Machine Learning-Based Missing Value Imputation Method for Clinical Datasets". In: *IAENG Transactions on Engineering Technologies: Special Volume of the World Congress on Engineering 2012*. Dordrecht: Springer Netherlands, 2013, pp. 245–257. DOI: 10.1007/978-94-007-6190-2_19.

[112] Benjamin Recht et al. "Do ImageNet Classifiers Generalize to ImageNet?" In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 5389–5400. URL: https://proceedings.mlr.press/v97/recht19a.html.

[113] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)*. May 2016.

[114] Cecil R. Reynolds, Robert A. Altmann, and Daniel N. Allen. "Reliability". In: *Mastering Modern Psychological Testing: Theory and Methods*. Cham: Springer International Publishing, 2021, pp. 133–184. ISBN: 978-3-030-59455-8. DOI: 10.1007/978-3-030-59455-8_4. URL: https://doi.org/10.1007/978-3-030-59455-8_4.

[115] 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*. New York, NY, USA: Association for Computing Machinery, 2016, pp. 1135–1144. ISBN: 9781450342322. DOI: 10.1145/2939672.2939778.

[116] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Anchors: High-Precision Model-Agnostic Explanations". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). DOI: 10.1609/aaai.v32i1.11491.

[117] Peter R Rijnbeek and Jan A Kors. "Finding a short and accurate decision rule in disjunctive normal form by exhaustive search". In: *Machine learning* 80.1 (2010), pp. 33–62. DOI: 10.1007/s10994-010-5168-9.

[118] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature Machine Intelligence* 1.5 (2019), pp. 206–215. DOI: 10.1038/s42256-019-0048-x.

[119] Cynthia Rudin et al. "Interpretable machine learning: Fundamental principles and 10 grand challenges". In: *Statistics Surveys* 16.none (2022), pp. 1–85. DOI: 10.1214/21-SS133.

[120] J SAE. "3016: 2021 taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles". In: *Society of Automotive Engineers* (2021).

[121] Wissam Salhab et al. "A Systematic Literature Review on AI Safety: Identifying Trends, Challenges, and Future Directions". In: *IEEE Access* 12 (2024), pp. 131762–131784. DOI: 10.1109/ACCESS.2024.3440647.

[122] Suchi Saria and Adarsh Subbaswamy. "Tutorial: safe and reliable machine learning". In: *arXiv preprint arXiv:1904.07204* (2019). DOI: 10.48550/arXiv.1904.07204.

[123] Iqbal H Sarker et al. "Multi-aspect rule-based AI: Methods, taxonomy, challenges and directions toward automation, intelligence and transparent cybersecurity modeling for critical infrastructures". In: *Internet of Things* (2024), p. 101110. DOI: 10.1016/j.iot.2024.101110.

[124] Thomas Schnake et al. "Higher-Order Explanations of Graph Neural Networks via Relevant Walks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), pp. 1–1. DOI: 10.1109/TPAMI.2021.3115452.

[125] Luigi Scorzato. "Reliability and Interpretability in Science and Deep Learning". In: *Minds and Machines* 34.3 (2024), p. 27. DOI: 10.1007/s11023-024-09682-0.

[126] Philip Sedgwick. "Pearson's correlation coefficient". In: *Bmj* 345 (2012).

[127] Zahra Sedighi Maman et al. "A data analytic framework for physical fatigue management using wearable sensors". In: *Expert Systems with Applications* 155 (2020), p. 113405. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2020.113405.

[128] Michele Segata et al. "Plexe: A platooning extension for Veins". In: *2014 IEEE Vehicular Networking Conference (VNC)*. IEEE. 2014, pp. 53–60. DOI: 10.1109/VNC.2014.7013309.

[129] Prerna Sethi and Sathya Alagiriswamy. "Association rule based similarity measures for the clustering of gene expression data". In: *The open medical informatics journal* 4 (2010), p. 63. DOI: 10.2174/1874431101004010063.

[130] Glenn Shafer and Vladimir Vovk. "A tutorial on conformal prediction". In: (2007). DOI: 10.48550/ARXIV.0706.3188. URL: https://arxiv.org/abs/0706.3188.

[131] *Smoke detection dataset*. URL: https://www.kaggle.com/datasets/deepcontractor/smoke-detection-dataset.

[132] Timo Speith. "A review of taxonomies of explainable artificial intelligence (XAI) methods". In: *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency*. 2022, pp. 2239–2250. DOI: 10.1145/3531146.3534639.

[133] Elham Tabassi. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. en. 2023-01-26 05:01:00 2023. DOI: https://doi.org/10.6028/NIST.AI.100-1. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=936225.

[134] Florian Tambon et al. "How to certify machine learning based safety-critical systems? A systematic literature review". In: *Automated Software Engineering* 29.2 (2022), p. 38. DOI: 10.1007/s10515-022-00337-x.

[135] D. Tax and R. Duin. "Support vector domain description". In: *Pattern Recognition Letters* 20 (1999), pp. 1191–1199. DOI: 10.1016/S0167-8655(99)00087-2.

[136] D. Tax and R. Duin. "Support vector domain description". In: *Machine Learning* 54 (2004), pp. 45–66. DOI: 10.1023/B:MACH.0000008084.60811.49.

[137] Richard Tomsett et al. "Rapid trust calibration through interpretable and uncertainty-aware AI". In: *Patterns* 1.4 (2020). DOI: 10.1016/j.patter.2020.100049.

[138] Dustin Tran et al. "Plex: Towards reliability using pretrained large model extensions". In: *arXiv preprint arXiv:2207.07411* (2022). DOI: 10.48550/arXiv.2207.07411.

[139] *Turbofan engine degradation simulation data set, https://www.kaggle.com/datasets/behrad3d/nasa-cmaps*. Accessed: Feb 2023.

[140] Ivan Vaccari et al. "MQTTset, a New Dataset for Machine Learning Techniques on MQTT". In: *Sensors* 20.22 (2020). ISSN: 1424-8220. DOI: 10.3390/s20226578.

[141] Juozas Vaicenavicius et al. "Evaluating model calibration in classification". In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Ed. by Kamalika Chaudhuri and Masashi Sugiyama. Vol. 89. Proceedings of Machine Learning Research. PMLR, 16–18 Apr 2019, pp. 3459–3467. DOI: https://proceedings.mlr.press/v89/vaicenavicius19a/vaicenavicius19a.pdf.

[142] Volodya Vovk, Alexander Gammerman, and Craig Saunders. "Machine-Learning Applications of Algorithmic Randomness". In: *Proceedings of the Sixteenth International Conference on Machine Learning*. ICML '99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 444–453. ISBN: 1558606122. DOI: https://eprints.soton.ac.uk/258960/1/Random_ICML99.pdf.

[143] Sandra Wachter, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR". In: *Harv. JL & Tech.* 31 (2017), p. 841.

[144] Warren E Walker et al. "Defining uncertainty: a conceptual basis for uncertainty management in model-based decision support". In: *Integrated assessment* 4.1 (2003), pp. 5–17. DOI: 10.1076/iaij.4.1.5.16466.

[145] Fulton Wang and Cynthia Rudin. "Falling Rule Lists". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Guy Lebanon and S. V. N. Vishwanathan. Vol. 38. Proceedings of Machine Learning Research. San Diego, California, USA: PMLR, Sept. 2015, pp. 1013–1022. URL: https://proceedings.mlr.press/v38/wang15a.html.

[146] Tong Wang. "Multi-value rule sets for interpretable classification with feature-efficient representations". In: *Advances in neural information processing systems* 31 (2018). URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/32bbf7b2bc4ed14eb1e9c2580056a989-Paper.pdf.

[147] Tong Wang and Qihang Lin. "Hybrid Predictive Models: When an Interpretable Model Collaborates with a Black-box Model". In: *Journal of Machine Learning Research* 22.137 (2021), pp. 1–38. URL: https://jmlr.org/papers/volume22/19-325/19-325.pdf.

[148] Tong Wang and Cynthia Rudin. "Causal rule sets for identifying subgroups with enhanced treatment effects". In: *INFORMS Journal on Computing* (2022). DOI: 10.1287/ijoc.2021.1143.

[149] Tong Wang et al. "A bayesian framework for learning rule sets for interpretable classification". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 2357–2393.

[150] Dennis Wei et al. "Generalized linear rule models". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 6687–6696. URL: https://proceedings.mlr.press/v97/wei19a/wei19a.pdf.

[151] Melissa Whatley. "One-Way ANOVA and the chi-square test of independence". In: *Introduction to Quantitative Analysis for International Educators*. Springer, 2022, pp. 57–74.

[152] White House Office of Science and Technology Policy. *Blueprint for an AI Bill of Rights*. The White House. Accessed: 2024-10-22. 2022. URL: https://www.whitehouse.gov/ostp/ai-bill-of-rights/.

[153] Nerys Williams. "The Borg Rating of Perceived Exertion (RPE) scale". In: *Occupational Medicine* 67.5 (July 2017), pp. 404–405. ISSN: 0962-7480. DOI: 10.1093/occmed/kqx063.

[154] Michael M Wolf. "Mathematical foundations of supervised learning". In: (2023). URL: https://mediatum.ub.tum.de/doc/1723378/1723378.pdf.

[155] Ning Xiong. "Fuzzy rule-based similarity model enables learning from small case bases". In: *Applied Soft Computing* 13.4 (2013), pp. 2057–2064. DOI: 10.1016/j.asoc.2012.11.009.

[156] Jiancheng Yang et al. *MedMNIST v2-A large-scale lightweight benchmark for 2D and 3D biomedical image classification*. Vol. 10. 1. Nature Publishing Group UK London, 2023, p. 41.

[157] Jingkang Yang et al. "Generalized out-of-distribution detection: A survey". In: *International Journal of Computer Vision* (2024), pp. 1–28. DOI: 10.1007/s11263-024-02117-4.

[158] Won Keun Youn et al. "Software certification of safety-critical avionic systems: DO-178C and its impacts". In: *IEEE Aerospace and Electronic Systems Magazine* 30.4 (2015), pp. 4–13. DOI: 10.1109/MAES.2014.140109.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was pdfLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete TEX-system installation.