Ensuring Safe Social Navigation via Explainable Probabilistic and Conformal Safety Regions

Sara Narteni[1,2][0000−0002−0579−647X], Alberto Carlevaro[1,4][0000−0002−7206−5511], Jérôme Guzzi[3][0000−0002−1263−4110], and Maurizio Mongelli[1]

[1] CNR-IEIIT, Corso F.M. Perrone 24, 16152, Genoa, Italy
name.surname@ieiit.cnr.it
[2] Politecnico di Torino, DAUIN Department, 10129, Turin, Italy
[3] Dalle Molle Institute for Artificial Intelligence (IDSIA), USI-SUPSI, Lugano, Switzerland
[4] Funded Research Department, Aitek SpA, Via della Crocetta 15, Genoa, 16122, Italy

Abstract. The recent advancements of Artificial Intelligence (AI) have generated a lot of interest in the robotics community. Indeed, AI can find application in a wide variety of problems. Among these, social navigation of mobile robots is a big challenge, where ensuring non-harmful behaviors of the robotic system is fundamental.
In this paper, we consider a simulated navigation problem that involves a fleet of mobile agents moving in a cross scenario, governed by a human-like behavior. With the purpose of avoiding collisions among them, we show how safe and explainable AI (XAI) methods can constitute useful tools to tailor the parameters of the behavior towards a safe, collision-free, navigation. We first explore how global native rule-based classification provides interpretable characterizations of the agents' behavior. Afterwards, we derive safety regions, $\mathcal{S}_\varepsilon$, denoting the zones in the parameters space where collisions are avoided, with a maximum error given by ε. The design of the regions is based on scalable classifiers, a technique to tune the decision function of a machine learning (ML) classifier so to bound its error on a desired class to a predefined level, combined with either probabilistic scaling (probabilistic safety regions, PSR), or with conformal prediction theory (conformal safety regions, CSR). Finally, we investigate how explainability can be provided to these regions by extracting local rules from their boundaries.

Keywords: safe navigation · robotics · probabilistic safety regions · conformal prediction · interpretability · rule-based models.

1 Introduction

Nowadays, machine and deep learning play a fundamental role in many disciplines, and robotics stands out as one of the fields where autonomous decision-making can find fertile ground. Artificial intelligence (AI) can indeed support a wide range of robotics applications, spanning from object detection or recognition to healthcare, manufacturing, agriculture, and many others [42]. Among

these, assistive robotics is certainly of great interest, with the high diffusion of social robots designed to provide assistance to people in daily-life tasks such as indoor and outdoor motion, interacting with them and their surrounding environment [30,41]. The development of sophisticated mobile robots [40], leveraging advanced sensors and the latest AI algorithms, supporting mapping, localization and navigation phases [6,47,50], helps boosting this field.
However, such a massive presence of automation elements makes social robotics a safety-critical scenario where it is of paramount importance to ensure that the considered AI system behaves as intended, without causing harm to its users and all the other people or things involved.
In this direction, research can benefit from both simulation and certification methods for the AI algorithms. Simulation can help providing insights to the scenarios of interest for the reliable design of AI-based solutions, collecting data that may be hard to achieve with real experimentation [7,46]. Besides good training data, AI models also need validation techniques, which is topic of safe AI research, a sub-category of the *trustworthy AI* paradigm [20]. This involves pursuing performance guarantees of the AI while taking into account the social environment where the AI acts. With respect to this, explainable artificial intelligence (XAI) is a fundamental element, providing transparency to the autonomous decision-making process [26].

In this work, we consider a simulated scenario of social navigation, inspired to human movement, where robotic agents move between pairs of opposing targets: in this setting, agents may collide each other while reaching their targets. Therefore, we propose the application of a technique that combines the notion of scalable classifiers [4] with rule-based XAI models [37] to drive the search of *explainable probabilistic and conformal safety regions*, in the simulation parameters space, where safe, collision-free, navigation is guaranteed with controllable high probability, and where XAI helps shedding light into the parameters values defining such regions.
The remaining of the paper is structured as follows: Section 2 reports the existing literature in the field; Section 3 describes the theory of scalable classifiers and how their definition, combined with methodologies from order statistics, leads to probabilistic safety regions and conformal safety regions; Section 4 presents the adopted rule-based classification models; Section 5 describes the simulation-based robotic navigation environment; Section 6 reports the performed experimentation and the obtained results; lastly, Section 7 concludes the paper.

2 Related Works

Evaluating the reliability and trustworthiness of AI-based cyber-physical systems is becoming an increasingly essential requirement in modern engineering. This is because AI plays a critical role in the cyber-physical systems of everyday life, with applications including strategic infrastructure, such as energy industry [43] or construction sector [10], medical devices [49], computer vision [29], large language models [25] and especially trustworthy autonomous driving that

is becoming of pivotal importance for many political entities like the European Union [8]. Among such a wide range of topics, the AI community has a growing interest in the field of robotics and the use of reinforcement learning (RL) algorithms in it with safety-critical applications such as drone reload planning as in [45]. More in depth, the problem of guaranteeing a safe robotic behavior (e.g., collision-free navigation) has been addressed by both the model-driven world of control theory and the data-driven one of machine learning (especially RL): the combination of the two offer a great opportunity to design safe regions (i.e., the states where robots can safely operate) characterized by strong guarantees of the model-driven component and the generalization ability to unseen contexts of the RL part [3]. Techniques in this direction include RL solutions towards safety and robustness, such as safe exploration and optimization [32,24], and uncertainty-aware RL [48]. Collision avoidance is one of the fundamental tasks in smart mobility [17], and is also very important in robotic navigation, where safe RL finds fertile ground. For example, in [13,12] deep reinforcement learning solutions for safe and efficient navigation in complex crowded scenarios are presented. In [23] an uncertainty-aware model-based learning algorithm that estimates the probability of collision together with a statistical estimate of uncertainty is proposed. In [28] MCDropout and Bootstrapping techniques are used within Long-Short-Term-Memory (LSTM) models of a RL framework, to give computationally efficient uncertainty estimates, with application to crowded environments.

Deep Reinforcement Learning is not the only way to collision avoidance in robotics. Authors of [27] propose a method, Probabilistic Safety Barrier Certificates (PrSBC) using Control Barrier Functions, to individuate, in closed-form, the space of admissible control actions that are probabilistically safe with provable theoretical guarantees. Gaussian Mixture Models/Regression are used in [21] to develop an adaptive obstacle avoidance approach for collaborative robots that is able to adapt in order to maintain a safety distance between the robots. Also, examples of safe navigation that only relies on sensors data are present in literature, like the work in [38], where autonomous and safe robot navigation is based on a method of curvature trajectory control from a Lidar sensor, able to keep a safety distance. In addition, the proposed algorithm can avoid dynamic obstacles while smoothing the robot's trajectories.

In the landscape of trustworthy AI approaches for robotics, another key point is the role of XAI. Indeed, explainability and interpretability are an essential ingredient to increase users' trust in the intelligent system, thus favoring its adoption in everyday life [11]. The work proposed in [44] studies and compares a RL-based approach with a fuzzy logic system for risk mitigation of robotics system in smart manufacturing; the latter method consists of linguistic rules that were manually defined according to existing standards for collaborative robots. Also, in [51] a rule-based RL approach for robotic navigation is presented, where rules are built to reduce the redundant exploration space and guide the exploration strategy, and are integrated in a RL model as a form of external

knowledge. However, these rules are not learned automatically from the data, via white-box machine learning models.
To the best of our knowledge, what we propose in this work makes a step beyond the existing research, by addressing the problem of collision avoidance through confidence guarantees of ML models other than deep/reinforcement learning ones, and with the objective of providing interpretable indications for safe navigation.

3 Confidence Regions for Machine Learning

In the following sections we introduce the concept of "safety region", conceived as the subset of the input space where probabilistic guarantees on the prediction of the target (safe) class are given. The section starts with the introduction of a special class of classifiers, namely the *scalable classifiers* (Section 3.1), that rely on the idea of deforming the classifier's boundary according to a tunable scalar parameter. The concept of scalable classifier is then used together with two statistical methodologies, probabilistic scaling (Section 3.2) and conformal prediction (Section 3.3), to construct the desired safety regions.

3.1 Scalable Classifiers

Given an input space $\mathcal{X} \subseteq \mathbb{R}^d$, $d \in \mathbb{N}^+$, and an output space $\mathcal{Y} = \{-1, +1\}$, a (binary) Scalable Classifier [4] is a classifier formulated as follows

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

where the function $f_{\boldsymbol{\theta}} : \mathcal{X} \times \mathbb{R} \longrightarrow \mathbb{R}$ is the so-called *classifier predictor*. We emphasize the dependency of the classifier from tunable hyperparameters $\boldsymbol{\theta}$ for which, obviously, a different choice of them can correspond to a very different classifier. The addition of the scalar parameter ρ plays a central role to define a controllable and reliable framework for classification: this parameter can be changed in order to adjust the boundary of the classification for satisfying predefined requirements on the evaluation metrics of the classification (as controlling the number of false positives). The classifier predictor $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho)$ has to satisfy some assumptions, as continuity and being monotonically increasing (see Assumption 1 of [4]). It is worth noting that any (binary) classifier, $\hat{f}(\boldsymbol{x})$ can be made scalable just adding the scalar parameter in an additive way, i.e.

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

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 learned weights, φ is a feature map and b is the offset. Its scalable version is easily obtained adding ρ additively, i.e. $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) = \boldsymbol{w}^\top \varphi(\boldsymbol{x}) - b + \rho$. This is valuable since also neural network output

can be easily controlled by a rescaling of this additional parameter, without retraining. Finally, we introduce $\bar{\rho}(\boldsymbol{x})$ to describe how much it is necessary to vary f such that the point $\boldsymbol{x}$ lies exactly on the border of the classifier:

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

Given the idea that the class $+1$ refers to "safety" and -1 to "unsafety", the introduction of Scalable Classifiers allows to define another important concept to assess safety in classification, the so-called $\rho-$*safe set*:

$$\mathcal{S}(\rho) = \{\, \boldsymbol{x} \in \mathcal{X} \;:\; f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) < 0 \,\}. \tag{2}$$

Then, $\mathcal{S}(\rho)$ is the region where the classifier predicts $+1$, or, in other words, where the classification is supposed to be safe. However $\mathcal{S}(\rho)$ itself does not provide any probabilistic guarantee on the actual result of the classification, but it only defines a region where the classifier predicts $+1$ without taking into account misclassification error. The aim is to obtain a *probabilistic* safety region $\mathcal{S}_\varepsilon$ that with a probability no smaller than $1-\delta$ satisfies the probability constraint

$$\Pr\Big\{ y = -1 \text{ and } \boldsymbol{x} \in \mathcal{S}_\varepsilon \Big\} \leq \varepsilon. \tag{3}$$

This can be achieved through a simple procedure relying on a calibration set $\mathcal{Z}_c \doteq \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c} \subset \mathcal{X} \times \mathcal{Y}$ and techniques belonging to the field of order statistics: probabilistic scaling [31] and conformal prediction [1]. It has to be remarked that the set $\mathcal{S}_\varepsilon$ is supposed to be nonempty, otherwise (3) would be trivially satisfied. Specifically, as it would be clear from the next section, a special value of ρ, namely ρ_ε, that defines $\mathcal{S}_\varepsilon$ is always provided by construction. However, it is possible that $\mathcal{S}_\varepsilon \cap \mathcal{X}$ is empty, due to various reasons like a bad pre-trained classifier (insufficient number of training data, bad hyperparameter choices, non-separability of the data etc. etc.) or simply the fact that the classifier cannot confidently predict the class $+1$ while satisfying the probabilistic constraint in (3). This might suggest that the classification problem to be addressed is inherently ill-conditioned or that the solution is poorly designed.

3.2 Probabilistic Scaling

Before introducing 1, that proves (3) is achievable, it is necessary to define the concept of *generalized max*:

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

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

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

Moreover, to better introduce the concept of "scalable classifier" we make the following assumptions:

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

$$\rho_1 > \rho_2 \;\Rightarrow f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho_1) > f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho_2), \quad \forall \boldsymbol{x} \in \mathcal{X}. \tag{4}$$

We assume also that

$$\lim_{\rho \to -\infty} f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho) < 0 < \lim_{\rho \to \infty} f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho), \quad \forall \boldsymbol{x} \in \mathcal{X}. \tag{5}$$

Then, we can introduce:

Theorem 1 (Probabilistic Safety Region, [4]). *Consider the classifier* (1), *and suppose that Assumption 1 holds and that* $\Pr\{\boldsymbol{x} \in \mathcal{X}\} = 1$. *Suppose that* $\delta \in (0,1)$, $\varepsilon \in (0,1)$ *and take a calibration set* $\mathcal{Z}_c \doteq \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c}$ (n_c *i.i.d. samples) and an integer parameter* $1 \leq r \leq n_c$ *such that*

$$n_c \geq \frac{7.47}{\varepsilon} \ln \frac{1}{\delta}, \qquad r = \left\lceil \frac{\varepsilon n_c}{2} \right\rceil.$$

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

$$\rho_\varepsilon \doteq \max^{(r)} \left(\{\bar{\rho}(\tilde{\boldsymbol{x}}_j^U)\}_{j=1}^{n_U}\right), \tag{6}$$

where $\bar{\rho}(\tilde{\boldsymbol{x}}_j^U)$ *is such that* $f(\tilde{\boldsymbol{x}}_j^U, \bar{\rho}(\tilde{\boldsymbol{x}}_j^U)) = 0$ *for all* $j \in [n_U]$. *Define then the corresponding* ρ_ε*-safe set*

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

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

$$\Pr\Big\{y = -1 \text{ and } \boldsymbol{x} \in \mathcal{S}_\varepsilon\Big\} \leq \varepsilon. \tag{7}$$

The proof is available in [4]. Thus, probabilistic scaling provides a way to build safety regions simply scaling the offset of the classifier according to the number of negative samples of a calibration set, regardless any assumption on the probability distribution of the data and the classifier chosen.

3.3 Conformal Prediction

Conformal Prediction is a relatively new framework developed starting in the late nineties and early two thousand by V. Vovk. We refer the reader to the surveys [2,15] for a gentle introduction to this methodology. CP is mainly an a-posteriori verification of the designed classifier, and in practice returns a measure of its "conformity" to the calibration data. We consider the particular implementation of CP discussed in [2], relative to the so-called "inductive" CP: in this setting, starting from a given predictor and a calibration set $\mathcal{Z}_c$, CP allows to construct a

new predictor with given probabilistic guarantees on the basis of a *score function* $s : \mathcal{X} \times \mathcal{Y} \longrightarrow \mathbb{R}$ that encodes the agreement between a sample $\boldsymbol{x}$ and a candidate label $\hat{y}$. The larger is the score the worse is the agreement between $\boldsymbol{x}$ and $\hat{y}$. Scalable classifiers give a natural definition of score function [5], based on their own classifier predictor:

$$s(\boldsymbol{x}, \hat{y}) = -\hat{y}\bar{\rho}(\boldsymbol{x}) \tag{8}$$

with $\bar{\rho}(\boldsymbol{x})$ such that $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \bar{\rho}(\boldsymbol{x})) = 0$. For example, the score function for the scalable SVM is $s(\boldsymbol{x}, \hat{y}) = -\hat{y}(b - \boldsymbol{w}^\top \varphi(\boldsymbol{x}))$. Computed then the $(\lceil (n_c + 1)(1 - \varepsilon) \rceil / n_c)$-quantile of the score values obtained on the calibration set, to every point $\boldsymbol{x}$, CP associates a set of "plausible labels"

$$C_\varepsilon(\boldsymbol{x}) = \{\ \hat{y} \in \{-1, 1\}\ :\ s(\boldsymbol{x}, \hat{y}) \leq s_\varepsilon\ \}$$

that, given any (unseen before) observation $(\tilde{\boldsymbol{x}}, \tilde{y})$, satisfies the following *marginal coverage* property:

$$\Pr\{\tilde{y} \in C_\varepsilon(\tilde{\boldsymbol{x}})\} \geq 1 - \varepsilon. \tag{9}$$

This formulation is oriented on the output set, but it can be adapted to give guarantees on the input set. In this regard, we introduce the following set

$$\Sigma_\varepsilon = \{\ \boldsymbol{x} \in \mathcal{X}\ :\ s(\boldsymbol{x}, +1) \leq s_\varepsilon,\ s(\boldsymbol{x}, -1) > s_\varepsilon\ \}. \tag{10}$$

that is the piece-wise set of the input sample $\boldsymbol{x}$ that have the marginal coverage probability to have as true label $y = +1$. We will refer to this set as *Conformal Safety Region* (CSR). With this formulation, to achieve (3) the following thorem holds.

Theorem 2 (Conformal Safety Region, [5]). *Consider the classifier* (1) *and suppose that Assumption 1 holds and that* $\Pr\{\boldsymbol{x} \in \mathcal{X}\} = 1$. *Consider then a calibration set* $\mathcal{Z}_c = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_c}$ *(n_c exchangeable samples), a level of error* $\varepsilon \in (0, 1)$, *a score function* $s : \mathcal{X} \times \mathcal{Y} \longrightarrow \mathbb{R}$ *as in* (8) *with the* $\lceil (n_c+1)(1-\varepsilon) \rceil / n_c$*-quantile* s_ε *computed on the calibration set. Define the* conformal scaling of level ε *as follows*

$$\rho_\varepsilon = |s_\varepsilon|,$$

and define the corresponding ρ_ε*-safe set*

$$\mathcal{S}_\varepsilon = \mathcal{S}(\rho_\varepsilon). \tag{11}$$

Then, given the conformal safety region of level ε, Σ_ε, *we have*

i) $\mathcal{S}_\varepsilon \subseteq \Sigma_\varepsilon$.
ii) $\mathcal{S}_\varepsilon = \Sigma_\varepsilon$ *if* $s_\varepsilon \leq 0$.

that is, $\mathcal{S}_\varepsilon$ *is a CSR. Moreover, it can be stated that*

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

The proof is available in [5].

Remark 1. In the following, we will denote the safety region $\mathcal{S}_\varepsilon$ obtained with the two methods with $\mathcal{S}_\varepsilon^{PS}$ for the probabilistic scaling approach (Theorem 1) and with $\mathcal{S}_\varepsilon^{CP}$ for the conformal prediction approach (Theorem 2).

4 Rule-based classification

Different techniques can lead to generate intelligible rules: depending on whether rules are learnt with the aim of explaining the whole dataset, or they are designed to provide interpretation to point predictions of a pre-trained ML classifier, two categories of rule extraction can be distinguished, following the common categorization of XAI [19,9]. In the first case, we deal with *global native rule generation*, while in the latter we speak about *local post-hoc rule extraction*.
The fundamental notation is the same regardless the specific kind of algorithm. Rule-based classifiers are machine learning models that provide their outcomes as sets of decision rules, i.e., rulesets. Independently on the specific way rules are learnt, a decision rule r is expressed with the following general syntax [33]:

$$\textbf{if } premise(r) \textbf{ then } consequence(r)$$

The *premise* part is also known as the *antecedent*, and states all the conditions on the input features that must be simultaneously met to make the rule satisfied, while the *consequence* part indicates the target class predicted by the rule.

4.1 Global native rule generation

Global rule-based classification involves learning a set of rules whose aim is to represent the whole logic of the model, being thus appropriate to be used on any data sample. Also, being *native* methods means that the learning mechanisms directly provides the rules, with no intermediary step. The problem can be formalized as follows.

Let us consider a set of inputs $\mathcal{T} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^{N} \in \mathcal{X} \times \mathcal{Y}$, with $\boldsymbol{x}_j \in \mathbb{R}^D$ and $y \in \{-1, +1\}$. A rule-based classifier trained on $\mathcal{T}$ generates a ruleset $\mathcal{R} = \{r_k\}_{k=1}^{M_r}$, with each rule r_k composed by a *premise* being expressed as the following conjunction:

$$premise(r_k) = \bigwedge_{i_k=1_k}^{N_k} c_{i_k}.$$

Each condition c_{i_k} corresponds to an interval on the input features, which can be bounded, only left-bounded or only right-bounded.
Further, rule *consequence* is expressed as:

$$consequence(r_k) = \hat{y}_k \in \{-1, +1\}$$

The performance of each rule r_k of the model can be measured by two metrics, the covering $C(r_k)$ and error $E(r_k)$ (commonly known as True Positive Rate and False Positive Rate of the rule, respectively), defined as follows:

$$C(r_k) = \frac{TP(r_k)}{TP(r_k) + FN(r_k)} \tag{12}$$

$$E(r_k) = \frac{FP(r_k)}{TN(r_k) + FP(r_k)} \tag{13}$$

where $TP(r_k), FP(r_k), TN(r_k), FN(r_k)$ are the true/false positives and true/false negatives associated to the classification of samples through rule r_k. Combined together, covering and error determine the *rule relevance*:

$$R(r_k) = C(r_k) \cdot (1 - E(r_k)) \tag{14}$$

Since they express the portion of points correctly covered by the rule, both covering and rule relevance can be considered as good metrics to evaluate how much the rule can generalize to unseen data.

Logic Learning Machine Logic Learning Machine (LLM) by Rulex [5] is a global rule-based classifier [35,36]based on three steps:i) *discretization* of the feature space and mapping to a Boolean lattice; ii) identification of groups of points associated to the output classes, through a *shadow clustering* method; finally, *rule generation* phase, where clusters are converted back to the original spaceand eventually combined into a set of intelligible rules.In this model, resulting rules can overlap (i.e., a sample may cover multiple rules).

In the inference phase, label assignment is performed by the LLM as follows. Let us denote with $\mathcal{R}^y_{\boldsymbol{x}}$ 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. Then, a class label $\hat{y}$ is assigned to $\boldsymbol{x}$ by solving the following problem [14]:

$$\hat{y} = \underset{y}{\operatorname{argmax}} \left(\frac{\sum_{r \in \mathcal{R}^y_{\boldsymbol{x}}} R(r)}{\sum_{r \in \mathcal{R}^y} R(r)} \right) \tag{15}$$

Hence, rule relevance has an important role in determining the inference results.

Skope Rules `skope-rules`[6] is a rule-based ML algorithm, inspired to RuleFit [16], that learns 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, a separate training is thus required for each class of the problem (if one wants to obtain rules for all classes).
The rule learning process is structured in three phases: i) a *bagging* of decision trees is trained and a decision rule is extracted from each path or sub-path of the ensemble; ii) the set of rules extracted from the bagging undergoes a *performance filtering* based on precision and recall thresholds, and, finally, iii) *semantic deduplication* further filters the rules to avoid redundant terms.
In practical applications, skope-rules model has been applied to several safety-critical classification tasks, as well as anomaly detection problems or cluster description [22].

[5] https://www.rulex.ai/
[6] https://github.com/scikit-learn-contrib/skope-rules

4.2 Local rule extraction via Anchors

Anchors [39] 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. 12.
Formally, an anchor A is defined as a set of predicates on input instance x to be explained, such that:

$$\Pr\{Prec(A) \geq \lambda_{prec}\} \geq 1 - \delta, \tag{16}$$

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

$$Prec(A) = \mathbb{E}_{D_x(z|A)}[\mathbb{1}_{f(x)=f(z)}] \tag{17}$$

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. The search of the optimal anchors is based on reinforcement learning [33] and can be expressed as the following combinatorial optimization problem:

$$\max_{A \text{ s.t. } \Pr\{Prec(A) \geq \lambda_{prec}\} \geq 1-\delta} C(A), \tag{18}$$

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

5 Simulation of Social Robotics Navigation

5.1 Navground simulator

The social navigation simulator Navground[7] allows to experiment with navigation algorithms. At its core, 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 circular 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. For this work, we simulate the scenario captured in Fig. 1, where four targets are located at the vertices of a cross. Agents go back and forth between pairs of points, crossing in the middle, using the navigation behavior described in the next section: one half of the agents between red/yellow and the other half between green/cyan.

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

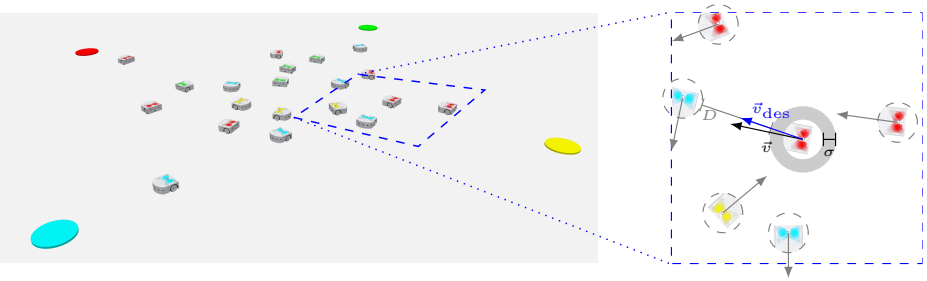

Fig. 1: Left: the simulated scenario where robots navigate back and forth between pairs of opposing targets (colored cylinders, distanced 4 m from each other); LEDs show the color of the current targets. Right: the navigation behavior for a robot (modelled as a disc with an additional safety margin σ) moving towards the red target: after selecting the desired direction considering its target and the state of its neighbors (modelled as discs), it computes desired velocity $\vec{v}_{\text{des}}$ taking into account the free distance D in that direction, and then modulates current velocity $\vec{v}$ towards $\vec{v}_{\text{des}}$.

5.2 Human-like Behavior

The Human-like behavior (HL, [18]) is a bio-inspired, conceptually simple and computationally light, local navigation algorithm, that extends and adapts to robotics a heuristic model for pedestrian motion [34]. 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. 1, the navigation behavior, at regular time-intervals Δt, performs the following steps 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: $\dot{v} = \frac{\vec{v}_{\text{des}} - \vec{v}}{\tau}$

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, we can define a value $\bar{\sigma}$ that ensures that no collision happen; for differential-drive robots, it is given by $\bar{\sigma} = 2v_{\text{opt}}(\Delta t + \tau + \tau_{\text{rot}})$, 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, we derive a more useful description for the

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

Table 1: The HL behavior's parameters we analyse in this work.

parameters region linked to safe trajectories. We summarize in Tab. 1 the HL parameters we are going to investigate.

6 Experimental Results on Collision Avoidance Task

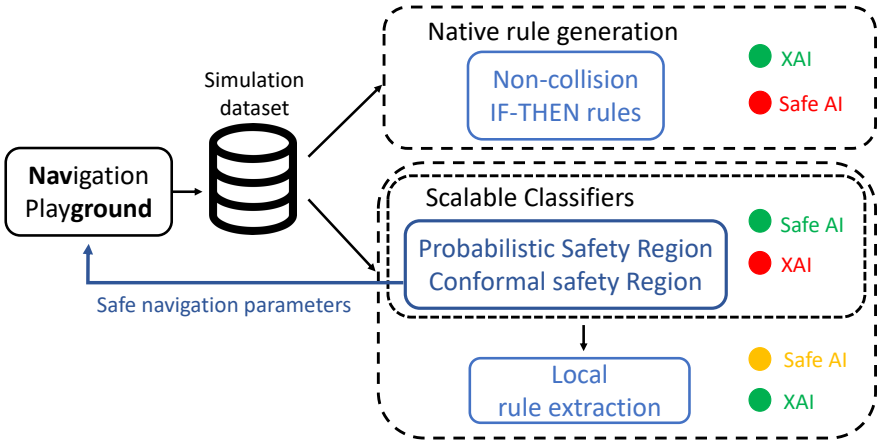

Fig. 2: Flowchart of the experimental methodology, involving XAI and safe AI (i.e., error control) components. Green, orange and red circles are used to denote a good, medium or bad level of the related component

In this Section, we present the experiments and results we carried out putting together the simulator and the illustrated techniques. The flowchart of Fig. 2

helps understanding the high-level workflow. After data collection through Navground simulator, we study the problem of collision avoidance guarantees in two different ways: on the one hand, the native rule generation solution is highly interpretable but does not guarantees any bound on the classification error; on the other hand, scalable classifiers through probabilistic and conformal safety regions are designed to provide probabilistic assurance on the error and thus can individuate simulation parameters ranges useful to guide the safe robotic navigation. Moreover, since safety regions lack of XAI features, we propose to locally extract rules from their boundaries to achieve a solution that brings a compromise provides interpretability again while maintaining a sufficient level of safety guarantee.

Code and data are made available for repeatability[8].

6.1 Data collection

Using Navground, we generated a suitable dataset to study the safety of robots' movement while avoiding collisions, via probabilistic scaling and conformal predictions methods, with interpretation via rule-based classifiers.

We executed $N = 10000$ simulation runs, each lasting 5 minutes of simulated time (i.e., a total simulated time of more than one month), with a group of 20 agents modelled after the Thymio robot,[9] 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 following parameters: i) $\Delta t = 0.1\,\mathrm{s}$; ii) $v_{\mathrm{opt}} = 0.12\,\mathrm{m/s}$; iii) $\tau_{\mathrm{rot}} = 0.5\,\mathrm{s}$; iv) σ sampled uniformly from $[0.0\,\mathrm{m}, 0.1\,\mathrm{m}]$; v) τ and η sampled uniformly from $[0.0\,\mathrm{s}, 1.0\,\mathrm{s}]$. For this configuration of parameters, the modelled upper bound required to ensure safety lies within the interval $[0.144\,\mathrm{m}, 0.384\,\mathrm{m}]$, where the two extrema correspond to $\tau = 0\,\mathrm{s}$ and $\tau = 1\,\mathrm{s}$ respectively.

For each simulation run, we recorded the value

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

which has been used by all simulated robots during that run, and the number of collisions that have happened. Then, we assigned a binary label y through the following criteria:

$$y = \begin{cases} +1 & \text{if number of collisions} = 0 \\ -1 & \text{if number of collisions} > 0 \end{cases} \tag{19}$$

Finally, we obtained the dataset $\mathcal{T}_{nav} = \{(\boldsymbol{x}_j, y_j) | j = 1, \ldots, N\}$, which we then analysed via reliable AI techniques.

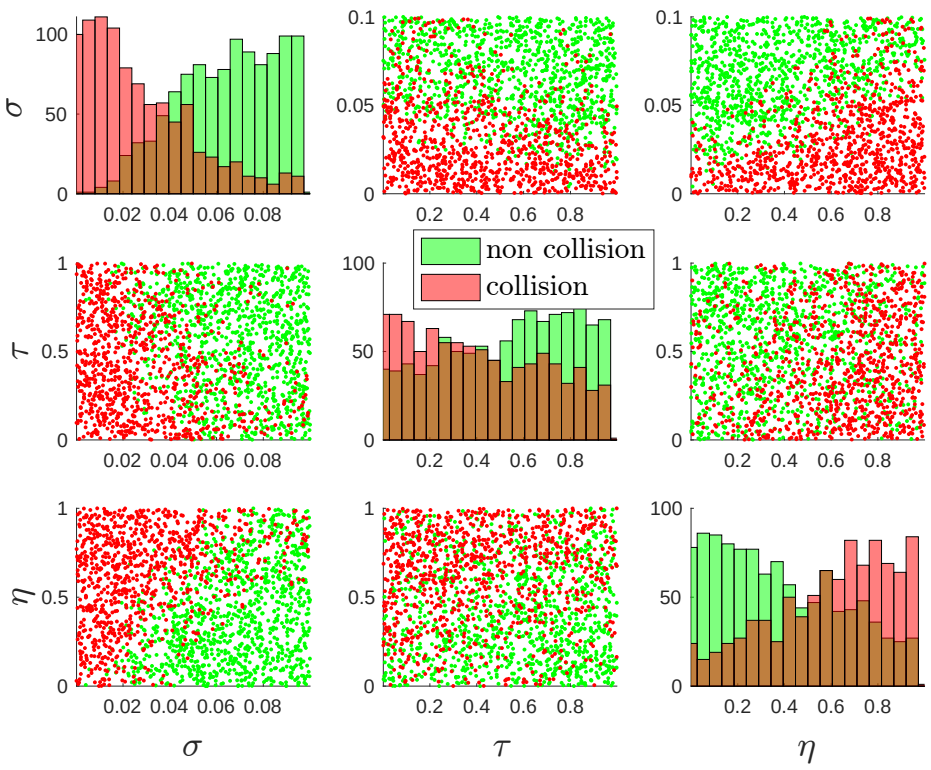

Fig. 3: Pairwise class distributions of the features in $\mathcal{T}_{nav}$. Green distribution refers to "non collision" label, red to "collision" label.

6.2 Data exploration

Before entering in the results in terms of confidence regions and explainability, a first visual inspection of how classes are distributed in the dataset is useful to understand the non-trivial nature of the problem.

Figure 3, along the top-left/bottom-right diagonal, shows the marginal class distributions of the three features, while the other plots are the features pairwise scatter plots. Despite the overall high superposition of red and green points, we can observe that, for larger σ values and lower τ or higher η, a region of non collision points can be individuated. The goal of the analyses detailed in the next sections is to characterize such a region in an interpretable way, either via direct decision rules generation or via post-hoc rule extraction from confidence regions.

[8] https://github.com/saranrt95/ExplainableSafetyRegions
[9] https://www.thymio.org

6.3 Native rule generation results

Two global rule-based models, the LLM and skope-rules, were adopted and compared to provide an intrinsically interpretable classification for the generated dataset.
Table 2 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.
Our labelling criterion (Eq. 19) assumes the non collision (+1) class as the positive one, and we remark that 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 our 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.

	# 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

Table 2: 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).

Examples of rules predicting *non collision* class are reported in Table 3 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.
The knowledge expressed by these rules confirms the visual information of Fig. 3. 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

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

Table 3: 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.

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 (see Section 6.1), 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 sections will be therefore dedicated to individuate *safety regions* where non collision class is predicted in high probability.

6.4 Safety regions via scalable SVM

The techniques detailed in Section 3 to derive the safety regions $\mathcal{S}_\varepsilon^{PS}$ (Theorem 1) and $\mathcal{S}_\varepsilon^{CP}$ (Theorem 2) were applied by adopting an SVM as classifier $\hat{f}_{\boldsymbol{\theta}}$. Specifically, we considered a 3rd degree polynomial kernel SVM with regularization parameter set to 0.3, and weighting of 0.5. This base model, prior to any error control, scored the 86% accuracy, 87.5% F1 score, 18% FPR, and 11% FNR, which is way similar to the LLM metrics, despite its more complex shape.

The design of the safety regions aims at bounding the rate of false positives, i.e., the collisions predicted as non collisions, to a desired ε value: to be able to maintain a large enough input space (i.e., finding regions $\mathcal{S}_\varepsilon^{PS}$ and $\mathcal{S}_\varepsilon^{CP}$ that do not reduce too much the space of parameters where the navigation can safely

operate), we chose to set it to $\varepsilon = 0.1$. With $\delta = 10^{-6}$, according to the formulas in Theorem 1, we then decided to set $n_c = 5000$ and, consequently, $r = 250$. Table 4 reports the performances for the safety regions obtained via probabilistic

	ρ_ε	**ACC**	**F1**	**TPR**	**FPR**	**FNR**	**TNR**
Probabilistic Safety Region	0.22	87.7	88.2	84	7	16	93
Conformal Safety Region	0.14	86.6	87.5	85	10	15	90

Table 4: Performance comparison between the adopted techniques for finding safety regions at $\varepsilon = 0.1$. The first column reports the optimal scaling parameter. 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).

scaling and conformal prediction techniques. First, we can note that all the metrics are very close in the two cases, as it is also pointed out by the similar values of the scaling parameter ρ_ε. We can observe that the FPR is lower than the ε bound in both cases, with $\mathcal{S}_\varepsilon^{PS}$ scoring even a lower error. Moreover, both methods manage to maintain a good balance with the FNR, thus being able to enclose high percentages of non collision points within the regions (TPR is 84% and 85% for $\mathcal{S}_\varepsilon^{PS}$ and $\mathcal{S}_\varepsilon^{CP}$, respectively). The blue and orange surfaces of Fig. 4 show the decision boundaries of the safety regions through PSR and CSR, respectively. The small difference between the values of ρ_ε reflects in the closeness between such surfaces, as well as the volumes enclosed by them.

6.5 Rule extraction from safety regions

Now, we want to derive interpretable approximations for both these regions, in the form of decision rules. Local rule extraction via Anchors (Sec. 4.2) was performed on a set of instances labelled as +1 by the scalable classifiers (i.e., being $f_{\boldsymbol{\theta}}(\boldsymbol{x}, \rho_\varepsilon) < 0$) and sampled at a small distance $d \leq 0.05$ from their border. Interestingly, such extraction converged to the same set of 4 rules, detailed in Table 5, for both probabilistic and conformal safety regions, and it is reasonable in light of the mentioned closeness of the scaling parameter. The low number of rules is a noticeable outcome too. Thanks to the mechanism of the Anchors algorithm, such a small set of rules performed quite well on the entire test set, even if extracted from a few points close to the decision boundary, thus approximating the safety regions while increasing interpretability.
More precisely, we assessed the anchors performance on the whole test set, by considering both the labels assigned via the scaling methods (either probabilistic or conformal) and the ground truth labels, as shown in Table 5. The first kind

of assessment allows to understand at what extent the generated anchors are faithful to the SVM-based safety regions, while testing with respect to ground truth is instead devoted to understand how such rules perform on the navigation problem. These rules confirm the visual intuitions from Fig. 3 as well as some of

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

Table 5: Anchors extracted from the scalable 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).

the discoveries from native XAI (Section 6.3), i.e., collisions are avoided by increasing safety margin and lowering τ. However, differing from native XAI where no error control was applied, the role of η becomes non-influential after Anchors rule extraction from the PSR and CSR, which suggests that this parameter is not relevant in profiling non-collision class in high probability.

The covering rate achieved with both scaling and ground truth outputs is satisfactory for all the four anchors, and this highlights that these rules manage to approximate sufficiently well the behavior of the scalable classifiers, even if their error is on average larger than the bound provided by the probabilistic and conformal scaling. This can be expected in light of their simplicity with respect to the polynomial shape of the scalable SVM. Nevertheless, reminding that the goal of the safety regions is to bound the false positive rate to 10% (i.e., we have $\varepsilon = 0.1$), we can observe that two of the four anchors are close to this bound too. Specifically, these are $\sigma > 0.07$ and $\tau \leq 0.25$, whose error[10] on the ground truth is 8.6% and 13%, respectively.

The light blue parallelepipeds of Fig. 4 show how these two rules are located in the feature space, with respect to the decision boundaries of the PSR (blue surface) and of the CSR (orange surface). The overall area delimited by the

[10] We remind that the error of a rule corresponds to its false positive rate, where the positive class is intended as the one predicted by the rule (see Eq. 13). Hence, in our case, the error refers to the percentage of collisions wrongly classified as non-collisions.

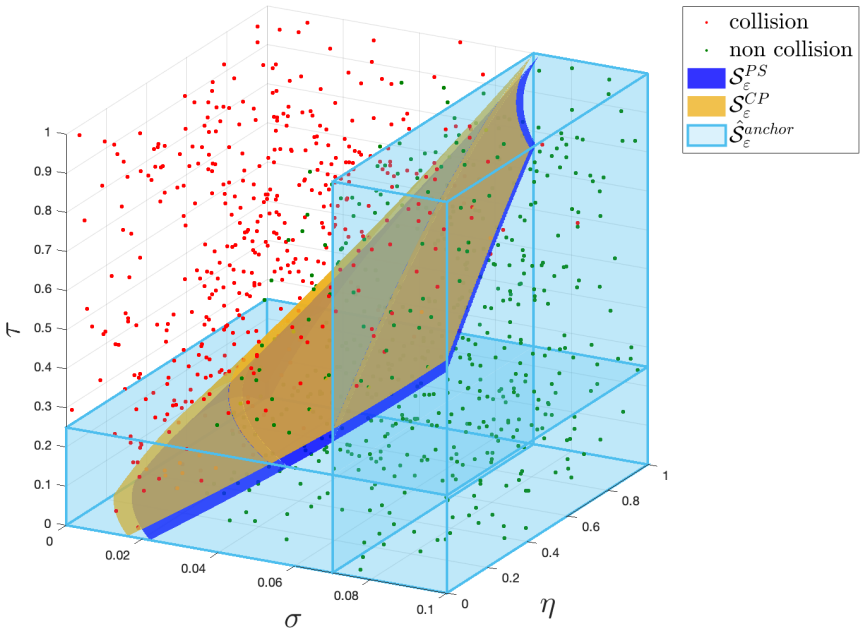

Fig. 4: Top-performing anchors extracted from the PSR ($\mathcal{S}_\varepsilon^{PS}$, area under the blue surface) and CSR ($\mathcal{S}_\varepsilon^{CP}$, area under the orange surface), at $\varepsilon = 0.1$

logical union of the anchors can be synthesised by the following region:

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

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). Considering the non trivial nature of the problem and of approximating a complex SVM shape via hyper-rectangular shapes (rules) while keeping the error bound as lower as possible, we can consider our results as a promising compromise between safety and trasparency. Indeed, this region 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.

7 Conclusions and Future Works

In this research, we presented a ML-based approach to safe, collision-free, mobile robots navigation. To achieve our goal, by leveraging simulation data from

NavGround simulator, we proposed the joint usage of safe and explainable AI techniques. Global rule-based classifiers (LLM and skope-rules) were first investigated to provide a first interpretable characterization of the simulation behavior. We then used a scalable SVM classifier and theories from order statistics to design safety regions $\mathcal{S}_\varepsilon$, where collisions are avoided with a predefined error level ε. More specifically, scalable SVM combined with probabilistic scaling allowed us to find probabilistic safety regions $\mathcal{S}_\varepsilon^{PS}$, while their combination with conformal prediction theory led to individuate conformal safety regions $\mathcal{S}_\varepsilon^{CP}$. Finally, local Anchor rules were extracted from these regions to give explainability. Overall, this method managed to find a good compromise between the more complex but more accurate equations of the safety regions and their simpler yet less precise rule-based version.

While in this work the focus was to provide safety guarantees, future extensions may include efficiency evaluations for deadlock avoidance. Also, different and more complex scenarios and/or behaviors may be explored, as well as adding other parameters to the analysis.

Acknowledgements

This work was partially supported by REXASI-PRO H-EU project, call HORIZON-CL4-2021-HUMAN-01-01, Grant agreement ID: 101070028.

References

1. Angelopoulos, A.N., Bates, S.: A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint arXiv:2107.07511 (2021)
2. Angelopoulos, A.N., Bates, S.: A gentle introduction to conformal prediction and distribution-free uncertainty quantification (2021). https://doi.org/10.48550/ARXIV.2107.07511, https://arxiv.org/abs/2107.07511
3. Brunke, L., Greeff, M., Hall, A.W., Yuan, Z., Zhou, S., Panerati, J., Schoellig, A.P.: Safe learning in robotics: From learning-based control to safe reinforcement learning. Annual Review of Control, Robotics, and Autonomous Systems **5**, 411–444 (2022)
4. Carlevaro, A., Alamo, T., Dabbene, F., Mongelli, M.: Probabilistic safety regions via finite families of scalable classifiers. arXiv preprint arXiv:2309.04627 (2023)
5. Carlevaro, A., Cantarero, T.A., Dabbene, F., Mongelli, M.: Conformal predictions for probabilistically robust scalable machine learning classification. arXiv preprint arXiv:2403.10368 (2024)
6. Cebollada, S., Payá, L., Flores, M., Peidró, A., Reinoso, O.: A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. Expert Systems with Applications **167**, 114195 (2021)
7. Choi, H., Crump, C., Duriez, C., Elmquist, A., Hager, G., Han, D., Hearl, F., Hodgins, J., Jain, A., Leve, F., et al.: On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. Proceedings of the National Academy of Sciences **118**(1), e1907856118 (2021)

8. D, F.L., E, G.G.: Trustworthy autonomous vehicles. Scientific analysis or review, Anticipation and foresight, Technical guidance KJ-NA-30942-EN-N (online), Luxembourg (Luxembourg) (2021). https://doi.org/10.2760/120385 (online)
9. Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G., et al.: Explainable ai (xai): Core ideas, techniques, and solutions. ACM Computing Surveys **55**(9), 1–33 (2023)
10. Emaminejad, N., Akhavian, R.: Trustworthy ai and robotics: Implications for the aec industry. Automation in Construction **139**, 104298 (2022). https://doi.org/https://doi.org/10.1016/j.autcon.2022.104298, https://www.sciencedirect.com/science/article/pii/S0926580522001716
11. Emaminejad, N., Akhavian, R.: Trustworthy ai and robotics: Implications for the aec industry. Automation in Construction **139**, 104298 (2022)
12. Everett, M., Chen, Y.F., How, J.P.: Collision avoidance in pedestrian-rich environments with deep reinforcement learning. IEEE Access **9**, 10357–10377 (2021). https://doi.org/10.1109/ACCESS.2021.3050338
13. Fan, T., Long, P., Liu, W., Pan, J.: Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. The International Journal of Robotics Research **39**(7), 856–892 (2020)
14. Ferrari, E., Verda, D., Pinna, N., Muselli, M.: 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. pp. 33–42. Springer (2022)
15. Fontana, M., Zeni, G., Vantini, S.: Conformal prediction: A unified review of theory and new challenges. Bernoulli **29**(1), 1 – 23 (2023). https://doi.org/10.3150/21-BEJ1447, https://doi.org/10.3150/21-BEJ1447
16. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles (2008)
17. Fu, Y., Li, C., Yu, F.R., Luan, T.H., Zhang, Y.: A survey of driving safety with sensing, vehicular communications, and artificial intelligence-based collision avoidance. IEEE transactions on intelligent transportation systems **23**(7), 6142–6163 (2021)
18. Guzzi, J., Giusti, A., Gambardella, L.M., Theraulaz, G., Di Caro, G.A.: Human-friendly robot navigation in dynamic environments. In: 2013 IEEE international conference on robotics and automation. pp. 423–430. IEEE (2013)
19. Hassija, V., Chamola, V., Mahapatra, A., Singal, A., Goel, D., Huang, K., Scardapane, S., Spinelli, I., Mahmud, M., Hussain, A.: Interpreting black-box models: a review on explainable artificial intelligence. Cognitive Computation **16**(1), 45–74 (2024)
20. High-Level Expert Group on AI: Ethics guidelines for trustworthy ai. Report, European Commission, Brussels (Apr 2019), https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai
21. Hu, Y., Wang, Y., Hu, K., Li, W.: Adaptive obstacle avoidance in path planning of collaborative robots for dynamic manufacturing. Journal of Intelligent Manufacturing **34**(2), 789–807 (2023)
22. https://2018.ds3-datascience-polytechnique.fr/wp-content/uploads/2018/06/DS3-309.pdf
23. Kahn, G., Villaflor, A., Pong, V., Abbeel, P., Levine, S.: Uncertainty-aware reinforcement learning for collision avoidance. arXiv preprint arXiv:1702.01182 (2017)
24. Kim, Y., Allmendinger, R., López-Ibáñez, M.: Safe learning and optimization techniques: Towards a survey of the state of the art. In: International Workshop on the

Foundations of Trustworthy AI Integrating Learning, Optimization and Reasoning. pp. 123–139. Springer (2020)
25. Liu, Y., Yao, Y., Ton, J.F., Zhang, X., Guo, R., Cheng, H., Klochkov, Y., Taufiq, M.F., Li, H.: Trustworthy llms: a survey and guideline for evaluating large language models' alignment (2023)
26. Longo, L., Brcic, M., Cabitza, F., Choi, J., Confalonieri, R., Del Ser, J., Guidotti, R., Hayashi, Y., Herrera, F., Holzinger, A., et al.: Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions. Information Fusion p. 102301 (2024)
27. Luo, W., Sun, W., Kapoor, A.: Multi-robot collision avoidance under uncertainty with probabilistic safety barrier certificates. Advances in Neural Information Processing Systems **33**, 372–383 (2020)
28. Lütjens, B., Everett, M., How, J.P.: Safe reinforcement learning with model uncertainty estimates. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 8662–8668. IEEE (2019)
29. Mackowiak, R., Ardizzone, L., Kothe, U., Rother, C.: Generative classifiers as a basis for trustworthy image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2971–2981 (2021)
30. Mahdi, H., Akgun, S.A., Saleh, S., Dautenhahn, K.: A survey on the design and evolution of social robots—past, present and future. Robotics and Autonomous Systems p. 104193 (2022)
31. Mammarella, M., Mirasierra, V., Lorenzen, M., Alamo, T., Dabbene, F.: Chance-constrained sets approximation: A probabilistic scaling approach. Automatica **137**, 110108 (2022)
32. Moldovan, T.M., Abbeel, P.: Safe exploration in markov decision processes. arXiv preprint arXiv:1205.4810 (2012)
33. Molnar, C.: Interpretable machine learning. Lulu. com (2020)
34. Moussaïd, M., Helbing, D., Theraulaz, G.: How simple rules determine pedestrian behavior and crowd disasters. Proceedings of the National Academy of Sciences **108**(17), 6884–6888 (2011)
35. Muselli, M.: Switching neural networks: A new connectionist model for classification (01 2005). https://doi.org/10.1007/11731177_4
36. Narteni, S., Orani, V., Cambiaso, E., Rucco, M., Mongelli, M.: On the intersection of explainable and reliable ai for physical fatigue prediction. IEEE Access **10**, 76243–76260 (2022). https://doi.org/10.1109/ACCESS.2022.3191907
37. Narteni, S., Orani, V., Vaccari, I., Cambiaso, E., Mongelli, M.: Sensitivity of logic learning machine for reliability in safety-critical systems. IEEE Intelligent Systems **37**(5), 66–74 (2022)
38. Ravankar, A., Ravankar, A.A., Rawankar, A., Hoshino, Y.: Autonomous and safe navigation of mobile robots in vineyard with smooth collision avoidance. Agriculture **11**(10), 954 (2021)
39. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: AAAI Conference on Artificial Intelligence (AAAI) (2018)
40. Rubio, F., Valero, F., Llopis-Albert, C.: A review of mobile robots: Concepts, methods, theoretical framework, and applications. International Journal of Advanced Robotic Systems **16**(2), 1729881419839596 (2019)
41. Santos, N.B., Bavaresco, R.S., Tavares, J.E., Ramos, G.d.O., Barbosa, J.L.: A systematic mapping study of robotics in human care. Robotics and Autonomous Systems **144**, 103833 (2021)
42. Soori, M., Arezoo, B., Dastres, R.: Artificial intelligence, machine learning and deep learning in advanced robotics, a review. Cognitive Robotics (2023)

43. Sulaiman, A., Nagu, B., Kaur, G., Karuppaiah, P., Alshahrani, H., Reshan, M.S.A., AlYami, S., Shaikh, A.: Artificial intelligence-based secured power grid protocol for smart city. Sensors **23**(19), 8016 (2023). https://doi.org/10.3390/s23198016
44. Terra, A., Riaz, H., Raizer, K., Hata, A., Inam, R.: Safety vs. efficiency: Ai-based risk mitigation in collaborative robotics. In: 2020 6th International Conference on Control, Automation and Robotics (ICCAR). pp. 151–160 (2020). https://doi.org/10.1109/ICCAR49639.2020.9108037
45. Theile, M., Bayerlein, H., Caccamo, M., Sangiovanni-Vincentelli, A.L.: Learning to recharge: Uav coverage path planning through deep reinforcement learning (2023)
46. Von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., et al.: Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. IEEE Transactions on Knowledge and Data Engineering **35**(1), 614–633 (2021)
47. Xiao, X., Liu, B., Warnell, G., Stone, P.: Motion planning and control for mobile robot navigation using machine learning: a survey. Autonomous Robots **46**(5), 569–597 (2022)
48. Zhang, J., Cheung, B., Finn, C., Levine, S., Jayaraman, D.: Cautious adaptation for reinforcement learning in safety-critical settings. In: International Conference on Machine Learning. pp. 11055–11065. PMLR (2020)
49. Zhang, J., Zhang, Z.m.: Ethics and governance of trustworthy medical artificial intelligence. BMC Medical Informatics and Decision Making **23**(1), 7 (2023)
50. Zhu, K., Zhang, T.: Deep reinforcement learning based mobile robot navigation: A review. Tsinghua Science and Technology **26**(5), 674–691 (2021)
51. Zhu, Y., Wang, Z., Chen, C., Dong, D.: Rule-based reinforcement learning for efficient robot navigation with space reduction. IEEE/ASME Transactions on Mechatronics **27**(2), 846–857 (2022). https://doi.org/10.1109/TMECH.2021.3072675