Developing safe and explainable autonomous agents: from simulation to the real world

Federico Bianchi[1,†], Alberto Castellini[1,†], Alessandro Farinelli[1,†], Luca Marzari[1,†], Daniele Meli[1,*,†], Francesco Trotti[1,†] and Celeste Veronese[1,†]

[1]University of Verona, strada Le Grazie 15, 37135, Verona, Italy

Abstract

Responsible artificial intelligence is the next challenge of research to foster the deployment of autonomous systems in the real world. In this paper, we focus on safe and explainable design and deployment of autonomous agents, e.g., robots. In particular, we present our recent contributions to: *i)* safe and explainable planning, leveraging on safe Reinforcement Learning (RL) and neurosymbolic planning; *ii)* effective deployment of RL policies via model-based control; *iii)* formal verification of the safety of deep RL policies; and *iv)* explainable anomaly detection of complex real systems.

Keywords

Safe Reinforcement Learning, Formal verification of neural networks, Neurosymbolic AI, Planning under uncertainty

1. Introduction

Artificial Intelligence (AI) and robotics are pervading everyday activities, from industrial automation [1] to environmental monitoring [2]. As more and more sophisticated autonomous cognitive systems interact with humans in complex scenarios, the development of *responsible AI* solutions [3] becomes a fundamental design requirement, as prescribed also by the latest international regulations [1]. Responsible AI involves several aspects, including safety, transparency and trustability [4]. Safety regards providing guarantees about the behavior of AI systems, e.g., autonomous robotic systems, in terms of performance and potential harm to the surrounding environment or humans. Transparency and trustability are related to the perception of humans interacting with the AI system, e.g., the explainability and compliance of the system's behaviour to the expectation of humans from a moral or rational perspective [5].

In this paper, we summarize our main contributions in the field of responsible AI. We focus on autonomous agents, e.g., robots, and present our approach to responsible autonomy at different developmental stages. We first describe our solutions for safe and explainable planning in autonomous agents, via safe Reinforcement Learning (RL) and neurosymbolic approaches. We also analyze the problem of safe and compliant transfer of a planned policy on a physical robotic system, combining RL with model-based control. We then investigate how to provide formal guarantees of safety for black-box policies, e.g., from deep RL, via formal verification. Finally, we present solutions for efficient and explainable anomaly detection in autonomous systems.

Ital-IA 2024: 4th National Conference on Artificial Intelligence, organized by CINI, May 29-30, 2024, Naples, Italy

*Corresponding author.

† Authors are in alphabetical order.

daniele.meli@univr.it (D. Meli)

[1]European AI Act

2. Safe and explainable planning

We assume the autonomous agent and the environment are represented as a Markov Decision Process (MDP) $M = \langle S, A, T, R \rangle$, defining respectively the state space, the action space, the transition map, and the reward map. The first approach is based on Safe Policy Improvement (SPI) [6] and Monte Carlo Tree Search (MCTS) [7], which performs simulations in a model of the real environment to estimate the optimal policy *online*. The second solution combines MCTS with symbolic and logical reasoning, to guide the exploration of the RL agent towards better pathways.

2.1. Safe Policy Improvement with MCTS

Safe RL [9] investigates how to learn policies that maximize the performance of the agent, while respecting safety constraints during learning. One popular approach is Safe Policy Improvement with Baseline Bootstrapping (SPIBB) [10]. SPIBB starts from a baseline policy π_0 (e.g., a sub-optimal expert-designed policy). The algorithm then collects a batch dataset of trajectories (i.e., state-action pairs), and uses the baseline policy on less frequent state-action pairs. However, it does not scale to large state and action spaces.

To improve scalability, we recently introduced *Monte Carlo Tree Search Safe Policy Improvement with Baseline Bootstrapping* (MCTS-SPIBB) [8]. The algorithm exploits MCTS to estimate π_I *online*, hence it can scale to large domains, while keeping the asymptotic guarantees of convergence of SPIBB [8]. In [8] we compared MCTS-SPIBB

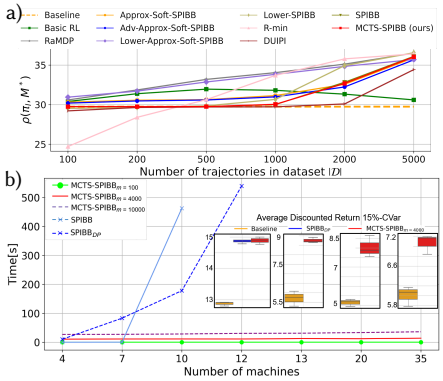


Figure 1: Safe Policy Improvement: a) Comparison of performance among SPI algorithms; b) Scalability comparison between MCTS-SPIBB and SPIBB [8].

with several state-of-the-art SPI algorithms on benchmark domains (see Figure 1.a). Furthermore, we showed that on very large state spaces, such as the standard SysAdmin benchmark[2] with up to 35 machines, MCTS-SPIBB is the only SPI algorithm capable of computing improved policies (see Figure 1.b).

2.2. Planning with logics in MCTS

MCTS may require a large number of online simulations when the state and action spaces are large. This becomes even more critical in Partially Observable MDPs (POMDPs), where part of the state is uncertain, hence a particle filter must be used to sample and estimate the actual state of the system, starting from a probability distribution called the belief. Recent online solvers for POMDPs, e.g., Partially Observable Monte Carlo Planning (POMCP) [11] and Determinized Sparse Partially Observable Trees (DESPOT) [12] require the definition of task-specific policy heuristics, in order to efficiently bias the exploration towards most fruitful policies. Moreover, it is essential to guarantee the exploration of only safe policies.

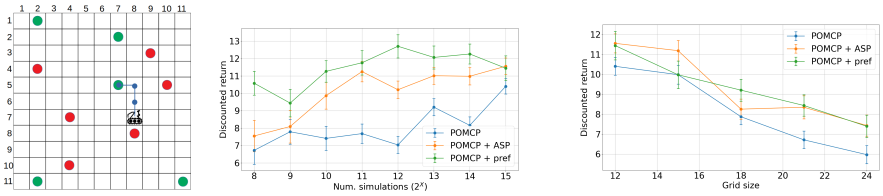
To this aim, in [13] we proposed an approach based on maximum satisfiability modulo theory [14] to probabilistically verify the adherence of the policy computed by POMCP to a set of user-defined specifications, expressed in a fragment of first-order logic. In this way, we can shield undesired actions in MCTS simulations, and increase the explainability of the generated policy thanks to the logic formalism. However, defining the logical policy specifications may be tedious and error-prone in realistic complex domains. For this reason, in [15, 16] we proposed an approach based on inductive logic programming [17] to learn logical policy heuristics from trajectories (belief-action pairs) of POMDP executions collected offline. Specifically, given a set of task-related concepts F provided by the user to describe the belief space, offline trajectories are converted to a logical formalism, where logical predicates encode concepts in F. As an example, consider the paradigmatic POMDP rocksample scenario depicted in Figure 2a, where a robotic agent must collect valuable rocks (green dots) avoiding worthless ones (red dots) in a grid world. The state of the POMDP includes information about the position of agents and rocks, and the probability (belief) of rocks to be valuable. The state can be translated to a logical representation in terms of the following concepts in F: the Manhattan distance `D` between the agent and each rock `R` `dist(R,D)` and the probability `P` of a rock `R` to be valuable `guess(R,P)`. Defining semantic concepts about the domain is easier than defining directly policy specifications, since it simply involves a re-interpretation of the state formalization.

We preliminarily learn policy specifications from trajectories collected from a rocksample agent operating in a 12×12 grid with 4 rocks. We adopt the logical formalism of Answer Set Programming (ASP) [18], which represents the state of the art for planning in first-order logic [19]. Our approach requires relatively few training trajectories (less than 800 in rocksample) to learn interpretable transparent policy specifications. Moreover, learned heuristics allow POMCP to use significantly fewer online simulations per step of execution (Figure 2b, achieving comparable performance with respect to expert-designed specifications (*pref*). Finally, the heuristics generalize to unseen problem instances, e.g., enhancing scalability to larger grid sizes (Figure 2c) which require a longer planning horizon, typically challenging for MCTS-based solvers. In [20], we also showed that this approach can be used to derive policy explanations of black-box model-free RL agents, in the context of autonomous driving.

3. Safe deployment in the real world

The policy computed by a RL-based planner, e.g., POMCP for POMDPs, cannot always be effectively and safely deployed on a real robotic system. Indeed, MCTS-based planners perform online simulations based on a model of the environment, but the chosen policy must be adapted to the inevitable unmodeled inaccuracies and non-linearities of the physical plant. To overcome this problem, in [21] we implemented the two-layer architecture depicted in Figure 3, combining a high-level controller based on POMCP with a low-level model-based controller: *The low-level controller* is designed using the inverse dynamics technique [22, 23], that allows to linearize via feedback the system. In particular, let

[2]SysAdmin: https://jair.org/index.php/jair/article/view/10341/24723



(a) (b) (c)

Figure 2: a) Rocksample setup; b) Results of [16] with few simulations and c) on larger grids.

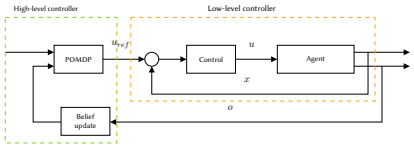


Figure 3: Block diagram of the two-layer architecture

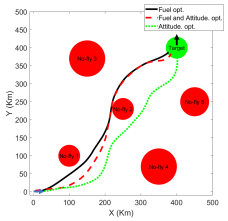
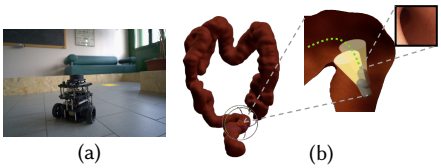


Figure 4: Drone paths. The black and blue arrows are, respectively, the desired yaw angle and drone initial yaw angle



(a) (b)

Figure 5: Realistic robotic scenarios: a) robotic mapless navigation; b) autonomous colonoscopy navigation.

$\dot{x} = f(x) + g(x)u$ be a non-linear dynamical system depending on state x and command action u. The controller is obtained as $u = \frac{1}{g(x)}(-f(x) + v)$, where v is an auxiliary control signal. Therefore, the low-level controller exploits the auxiliary control signal v, which is mapped as reference values for the controller, to compute the command u. *The high-level controller* is formalized as a POMDP that exploits the linearized closed-loop model to select the best local action u for the agents. In particular, the POMCP provides the sub-optimal reference values for the low-level controller optimizing user-defined objectives, encoded in the reward function. Note, the two-layer have different control loop sample rates; the low-level has to be fast since it has to provide the commands to the agent, while the high-level can be slower since it generates the reference values for the low-level.

The two-layer approach is tested in a scenario where an aerial drone has to reach a target area, avoiding some no-fly zones and minimizing fuel consumption or attitude error. Therefore, the reward function is composed of four contributions: an attractive potential component to reach the target, a repulsive component to avoid the no-fly zone, the fuel consumption and the heading error. The last two components are weighted to rank between different objectives. Figure 4 shows the trajectory followed by the drone optimizing only the fuel consumption (black line), both fuel and attitude (red dashed line) and only the attitude error (green dotted line). The black line follows the shortest path to minimize the fuel, the red line follows the shortest path but near the target position the attitude error component increases to align the drone with the desired attitude (black arrow). The green line follows the optimal path to minimize only the attitude.

4. Formal verification of deep RL

Trained RL policies, especially model-free RL policies encoded in a Deep Neural Network (DNN), do not guarantee to provably meet the safety standards required in the real world. For instance, DNNs are vulnerable to the so-called *adversarial inputs*, i.e., minimal input variations that fool the system to output an undesired value (or action) [24]. Consequently, in recent years, Formal Verification (FV) of DNNs (aka *DNN-Verification*) has been developed to provide formal guarantees on the behavior of these systems [25]. In particular, given a predefined safety property, the goal of *DNN-Verification* is to assert whether *at least one input configuration* exists that violates the property. However, given the non-convex and non-linear nature of DNNs, verifying safety properties in the worst case has been shown to be an NP-complete problem [26]. Moreover, the standard binary response of *DNN-verification* (safe vs. unsafe) does not provide sufficient information to compare the safety of different DNNs.

Table 1
Results of model selection. SAT indicates property violation. Θ's denote the safety property not to touch the colon wall in any cardinal direction.

	Safety Properties				
	$\Theta_{\uparrow}$	$\Theta_{\downarrow}$	$\Theta_{\leftarrow}$	$\Theta_{\rightarrow}$	**FV selection**
Method	SAT	SAT	SAT	SAT	*Safe models*
PPO	300	246	80	167	0
L-PPO	221	198	53	161	**3**

To overcome these limitations, in [27], we proposed a novel quantitative formulation of the *DNN-verification* problem, allowing to enumerate all unsafe regions for a given domain of interest and thus rank the models on the portion of unsafe regions they may have. However, we showed that this problem turns out to be #P-hard. Hence, in [28] we proposed ϵ-ProVe. Exploiting a controllable underestimation of the output reachable sets obtained via statistical prediction of tolerance limits [29], the algorithm provides a tight —with provable probabilistic guarantees— lower estimate of the (un)safe areas.

We validated *DNN-Verification* in realistic robotic safety-critical scenarios. In particular, in [30], we showed that *DNN-Verification* can be used to rank different successful DNN models according to the level of safety, verifying collision avoidance in robotic mapless navigation. We then applied a similar pipeline in a more safety-critical domain, namely autonomous colonoscopy navigation for colorectal detection with deep RL [31] (Figure 5). In particular, we trained an agent to navigate the endoscope in patient-specific colon models based on endoscopic images, using Constrained RL (CRL) to impose a safety cost for the agent to touch colon walls at the training stage. Nevertheless, due to the Lagrangian relaxation implemented by CRL to perform constrained optimization, safety may not be guaranteed. Hence, we adopted a model selection strategy that harnesses FV to evaluate the safety of a vast pool of trained policies to select the one the meets all the behavioral preferences specified. The results of our study are reported in Table 1 over 300 trained models, finding 3 completely safe models that provably meet the safety requirements.

Finally, to address the necessity of running the FV process only after training due to its computational complexity, in [32] we proposed an unconstrained DRL framework that leverages a novel sample-based method to approximate local violations of input-output conditions to foster the learning of safer behaviors inside the training loop. However, such conditions are typically hard-coded and require task-level knowledge, making their application intractable in challenging safety-critical tasks. To this end, in [33], we introduced the Collection and Refinement of Online Properties (CROP) framework to collect and refine safety properties during training. The combination of CROP with approximate violation inside the training loop allowed us to obtain a more robust approach with respect to other existing Safe DRL methodologies in the context of autonomous navigation, promoting safer behaviors while maintaining similar or better returns.

5. Explainable and data-efficient anomaly detection

Autonomous systems operating in the real world are required to reliably work over long periods of time (Long Term Autonomy, LTA) under changing and unpredictable environmental conditions. In this context, anomaly detection is crucial to promptly identify situations that diverge from the desired behaviour. Specifically, *unsupervised anomaly detection* aims to idenfity anomalies related to the global behavior of the system [34, 35, 36], monitoring multivariate time series generated from sensors and actuators and starting from the only knowledge of the *nominal* (i.e., anomaly-free) behavior.

We recently proposed two contributions in this area. namely, an online approach for detecting anomalous behaviors of robotic systems involved in complex LTA scenarios (HHAD) [37], and an adversarial data augmentation and retraining approach (HHAD-AUG) [38]. In HHAD [37], we use Hidden Markov Models (HMMs) to represent the nominal behavior of a robot. We then evaluate online the dissimilarity between the probability distribution of multivariate sensor time series in a sliding window and the emission probability of the related HMM hidden states. We adopt the Hellinger distance [39] as a distance measure since it is bounded (thus it lends itself to simpler interpretation and thresholding) and it is less noisy, hence more informative and discriminative.

In HHAD-AUG [38], we address the usual lack (or paucity) of anomalous examples and the noise that characterizes time series of real systems. We propose a data augmentation method based on perturbed (adversarial) time series [40], having the advantage of not requiring any prior knowledge about the application domain and data conformation. We generate adversarial examples only for nominal points, optimizing a loss function based on the Hellinger distance between the observed and the expected data distributions.

We evaluate our data augmentation and re-training approach on several public datasets, plus one collected from our aquatic drones developed in the EU H2020 project INTCATCH [41]. Results show that *(i)* the adversarial generation algorithms can generate meaningful adversarial examples for HHAD, employing them to significantly improve the performance of HHAD; *(ii)* our data augmentation method yields higher performance than examples generated by state-of-the-art augmentation methods; *(iii)* adversarial examples generated considering the Hellinger distance yield higher improvement than examples gen-

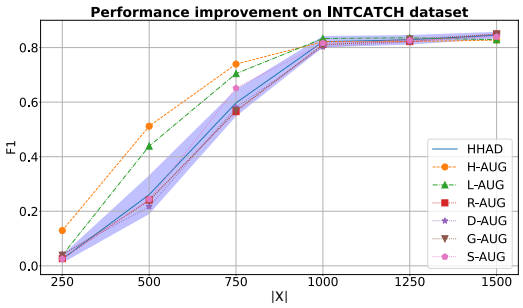


Figure 6: Average F1-score for the original detector HHAD and augmented detectors [38]: H-AUG (ours, based on Hellinger distance), L-AUG (ours, based on log-likelihood), R-AUG (random-based baseline), D-AUG (drift-based baseline), G-AUG (gaussian-based baseline), and S-AUG (SMOTE-based baseline) on different training set sizes in the INTCATCH dataset. Averages are computed over 30 datasets, for each dataset size.

erated considering standard log-likelihood; *(v)* the low computational complexity and high parallelizability of the proposed method allow for a fast data augmentation and retraining of HHAD. Figure 6 shows the results on the INTCATCH dataset [41].

Finally, we have recently addressed the problem of *explainable anomaly detection*, in order to provide useful information about the source of the anomaly for easier repair. To this aim, in [42] we showed that causal discovery based on Conditional Mutual Information (CMI) between time series can achieve higher performance than standard deep learning antomaly detectors, on a benchmark robotic dataset of the Pepper service robot[3]. Our methodology evaluates the variation of CMI between time series, thus providing a useful hint to the root cause of the anomaly. Moreover, it builds a nominal model of the real physical relations between variables of the system, thus resulting in higher robustness and more accurate anomaly detection, compared to DNN methods (95% vs 90 % F1-score and 100% precision).

6. Conclusion and future works

Our methodologies aim at increasing transparency and safety at different development levels, from planning to execution and verification. Our current research direction includes the online integration of symbolic learning and formal verification approaches into RL, focusing on the current scalability issues.

[3]https://sites.google.com/diag.uniroma1.it/robsec-data

References

[1] Z. Jan, F. Ahamed, W. Mayer, N. Patel, G. Grossmann, M. Stumptner, A. Kuusk, Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities, Expert Systems with Applications 216 (2023) 119456.

[2] M. Zuccotto, A. Castellini, D. L. Torre, L. Mola, A. Farinelli, Reinforcement learning applications in environmental sustainability: a review, Artificial Intelligence Review 57 (2024) 88.

[3] B. Shneiderman, Responsible ai: Bridging from ethics to practice, Communications of the ACM 64 (2021) 32–35.

[4] P. Mikalef, K. Conboy, J. E. Lundström, A. Popovič, Thinking responsibly about responsible AI and 'the dark side'of AI, 2022.

[5] I. Gabriel, Artificial intelligence, values, and alignment, Minds and machines 30 (2020) 411–437.

[6] P. Scholl, F. Dietrich, C. Otte, S. Udluft, Safe policy improvement approaches and their limitations, in: International Conference on Agents and Artificial Intelligence, Springer, 2022, pp. 74–98.

[7] M. Świechowski, K. Godlewski, B. Sawicki, J. Mańdziuk, Monte carlo tree search: A review of recent modifications and applications, Artificial Intelligence Review 56 (2023) 2497–2562.

[8] A. Castellini, F. Bianchi, E. Zorzi, T. D. Simão, A. Farinelli, M. T. J. Spaan, Scalable safe policy improvement via Monte Carlo tree search, in: Proceedings of the 40th International Conference on Machine Learning (ICML 2023), PMLR, 2023, pp. 3732–3756.

[9] R. Sutton, A. Barto, Reinforcement learning, An introduction, 2nd ed., MIT Press, 2018.

[10] R. Laroche, P. Trichelair, R. Tachet Des Combes, Safe policy improvement with baseline bootstrapping, in: Proceedings of the 36th International Conference on Machine Learning (ICML), PMLR, 2019, pp. 3652–3661.

[11] D. Silver, J. Veness, Monte-carlo planning in large pomdps, Advances in neural information processing systems 23 (2010).

[12] N. Ye, A. Somani, D. Hsu, W. S. Lee, Despot: Online pomdp planning with regularization, Journal of Artificial Intelligence Research 58 (2017) 231–266.

[13] G. Mazzi, A. Castellini, A. Farinelli, Risk-aware shielding of Partially Observable Monte Carlo Planning policies, Artificial Intelligence 324 (2023).

[14] C. Barrett, R. Sebastiani, S. A. Seshia, C. Tinelli, Satisfiability modulo theories, 2021.

[15] G. Mazzi, D. Meli, A. Castellini, A. Farinelli, Learning logic specifications for soft policy guidance in POMCP, in: Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent

Systems, AAMAS '23, IFAAMAS, 2023, pp. 373–381.
[16] D. Meli, A. Castellini, A. Farinelli, Learning logic specifications for policy guidance in POMDPs: an inductive logic programming approach, Journal of Artificial Intelligence Research 79 (2024) 725–776.
[17] S. Muggleton, L. De Raedt, Inductive logic programming: Theory and methods, The Journal of Logic Programming 19 (1994) 629–679.
[18] S. C. Tran, E. Pontelli, M. Balduccini, T. Schaub, Answer set planning: a survey, Theory and Practice of Logic Programming 23 (2023) 226–298.
[19] D. Meli, H. Nakawala, P. Fiorini, Logic programming for deliberative robotic task planning, Artificial Intelligence Review 56 (2023) 9011–9049.
[20] C. Veronese, D. Meli, F. Bistaffa, M. Rodríguez-Soto, A. Farinelli, J. A. Rodríguez-Aguilar, Inductive logic programming for transparent alignment with multiple moral values, in: CEUR WORKSHOP PROCEEDINGS, 2023, pp. 84–88.
[21] F. Trotti, A. Farinelli, R. Muradore, An online path planner based on pomdp for uavs, in: 2023 European Control Conference (ECC), IEEE, 2023.
[22] A. Isidori, Nonlinear control systems II, Springer, 2013.
[23] H. Khalil, Nonlinear Systems, Prentice Hall, 2002.
[24] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv:1312.6199 (2013).
[25] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, M. J. Kochenderfer, et al., Algorithms for verifying deep neural networks, Foundations and Trends® in Optimization 4 (2021) 244–404.
[26] G. Katz, C. Barrett, D. L. Dill, K. Julian, M. J. Kochenderfer, Reluplex: An efficient smt solver for verifying deep neural networks, in: International conference on computer aided verification, Springer, 2017, pp. 97–117.
[27] L. Marzari, D. Corsi, F. Cicalese, A. Farinelli, The #DNN-Verification Problem: Counting Unsafe Inputs for Deep Neural Networks, in: International Joint Conference on Artificial Intelligence (IJCAI), 2023, pp. 217–224.
[28] L. Marzari, D. Corsi, E. Marchesini, A. Farinelli, F. Cicalese, Enumerating safe regions in deep neural networks with provable probabilistic guarantees, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 2024, pp. 21387–21394.
[29] S. S. Wilks, Statistical prediction with special reference to the problem of tolerance limits, The annals of mathematical statistics 13 (1942) 400–409.
[30] G. Amir, D. Corsi, R. Yerushalmi, L. Marzari, D. Harel, A. Farinelli, G. Katz, Verifying learning-based robotic navigation systems, in: 29th International Conference TACAS, Springer, 2023, pp. 607–627.
[31] D. Corsi, L. Marzari, A. Pore, A. Farinelli, A. Casals, P. Fiorini, D. Dall'Alba, Constrained reinforcement learning and formal verification for safe colonoscopy navigation, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2023, pp. 10289–10294.
[32] E. Marchesini, L. Marzari, A. Farinelli, C. Amato, Safe deep reinforcement learning by verifying task-level properties, AAMAS '23, International Foundation for Autonomous Agents and Multiagent Systems, 2023, p. 1466–1475.
[33] L. Marzari, E. Marchesini, A. Farinelli, Online safety property collection and refinement for safe deep reinforcement learning in mapless navigation, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 7133–7139.
[34] A. Castellini, M. Bicego, F. Masillo, M. Zuccotto, A. Farinelli, Time series segmentation for state-model generation of autonomous aquatic drones: A systematic framework, Engineering Applications of Artificial Intelligence 90 (2020).
[35] A. Castellini, M. Bicego, D. Bloisi, J. Blum, F. Masillo, S. Peignier, A. Farinelli, Subspace clustering for situation assessment in aquatic drones: a sensitivity analysis for state-model improvement, Cybernetics and Systems 50 (2019) 658–671.
[36] A. Castellini, F. Masillo, M. Bicego, D. Bloisi, J. Blum, A. Farinelli, Subspace clustering for situation assessment in aquatic drones, in: Proc. 33th ACM/SIGAPP Symposium on Applied Computing, SAC, 2019, pp. 930–937.
[37] D. Azzalini, A. Castellini, M. Luperto, A. Farinelli, F. Amigoni, et al., HMMs for anomaly detection in autonomous robots, in: Proc. AAMAS, 2020, p. 105–113.
[38] A. Castellini, F. Masillo, D. Azzalini, F. Amigoni, A. Farinelli, Adversarial data augmentation for hmm-based anomaly detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (2023) 14131–14143.
[39] E. Hellinger, Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen, Journal für die reine und angewandte Mathematik 136 (1909) 210–271.
[40] F. Karim, S. Majumdar, H. Darabi, Adversarial attacks on time series, IEEE Trans Pattern Anal Mach Intell 43 (2020) 3309–3320.
[41] A. Castellini, D. Bloisi, J. Blum, F. Masillo, A. Farinelli, Multivariate sensor signals collected by aquatic drones involved in water monitoring: A complete dataset, Data Brief 30 (2020) 105436.
[42] D. Meli, Explainable online unsupervised anomaly detection for cyber-physical systems via causal discovery from time series, arXiv:2404.09871 (2024).