technologies

MDPI

Article

Defining a Metric-Driven Approach for Learning Hazardous Situations

Mario Fiorino [1,2], Muddasar Naeem [3,*], Mario Ciampi [2] and Antonio Coronato [3]

[1] Dipartimento di Automatica e Informatica (DAUIN), Politecnico di Torino, 10129 Turin, Italy; m.fiorino@unifortunato.eu
[2] Institute of High Performance Computing and Networking, National Research Council, 80133 Naples, Italy; mario.ciampi@icar.cnr.it
[3] Department of Computer Engineering, Università Giustino Fortunato, 82100 Benevento, Italy; a.coronato@unifortunato.eu
* Correspondence: m.naeem@unifortunato.eu

Abstract: Artificial intelligence has brought many innovations to our lives. At the same time, it is worth designing robust safety machine learning (ML) algorithms to obtain more benefits from technology. Reinforcement learning (RL) being an important ML method is largely applied in safety-centric scenarios. In such a situation, learning safety constraints are necessary to avoid undesired outcomes. Within the traditional RL paradigm, agents typically focus on identifying states associated with high rewards to maximize its long-term returns. This prioritization can lead to a neglect of potentially hazardous situations. Particularly, the exploration phase can pose significant risks, as it necessitates actions that may have unpredictable consequences. For instance, in autonomous driving applications, an RL agent might discover routes that yield high efficiency but fail to account for sudden hazardous conditions such as sharp turns or pedestrian crossings, potentially leading to catastrophic failures. Ensuring the safety of agents operating in unpredictable environments with potentially catastrophic failure states remains a critical challenge. This paper introduces a novel metric-driven approach aimed at containing risk in RL applications. Central to this approach are two developed indicators: the Hazard Indicator and the Risk Indicator. These metrics are designed to evaluate the safety of an environment by quantifying the likelihood of transitioning from safe states to failure states and assessing the associated risks. The fact that these indicators are characterized by a straightforward implementation, a highly generalizable probabilistic mathematical foundation, and a domain-independent nature makes them particularly interesting. To demonstrate their efficacy, we conducted experiments across various use cases, showcasing the feasibility of our proposed metrics. By enabling RL agents to effectively manage hazardous states, this approach paves the way for a more reliable and readily implementable RL in practical applications.

Keywords: safe artificial intelligence; reinforcement learning; hazard indicator; risk indicator

check for updates

Citation: Fiorino, M.; Naeem, M.; Ciampi, M.; Coronato, A. Defining a Metric-Driven Approach for Learning Hazardous Situations. *Technologies* **2024**, *12*, 103. https://doi.org/10.3390/technologies12070103

Academic Editor: Valeri Mladenov

Received: 27 May 2024
Revised: 19 June 2024
Accepted: 3 July 2024
Published: 4 July 2024

1. Introduction

Artificial intelligence (AI) and machine learning (ML) have dramatically improved human life by bringing a revolution in many areas [1]. AI-powered systems are integrated into everyday activities, from the technology embedded in smartphones to the personalized recommendations in online shopping. Although AI has brought many changes to the world, we need to develop robust safety ML algorithms to get the benefits of technology [2]. The algorithms trained with safety model can be utilized to monitor and guide online actions for safety compliance [3]. Creating an AI safety culture is essential in all workflows of ML development, from data gathering to product deployment. Reinforcement Learning (RL) is an important area of ML where an agent interacts with an unknown environment and tries to act in an optimal way by learning the dynamics of that environment [4] . Therefore, the safety aspect of learning about RL agents is more important.

RL, being a useful AI technology, is used in diverse domains [5]. RL algorithms have significant application in many areas including healthcare [6], UAVs, 5G and autonomous control [7], gesture classification [8], risk management [9], communication [10], and clinical assistance [11,12]. The increasing use of RL in many areas is a reflection of its exceptional ability to address complex problems that are sometime intractable for conventional ML techniques. However, the universal suitability of RL calls for a multi-faceted research mechanism, confining real-world validations, safety measures, and algorithmic improvements [13]. Researchers need to address questions like: are rewards on their own enough for the optimality of an RL model?

Although rewards are the basic metric in RL setup, an overemphasis on accumulated rewards may result in a misleading solution or evaluation. Therefore, this research work focuses on the definition of new metrics to manage hazardous situations or states alongside optimizing rewards.

This field is characterized by numerous challenges, including uncertainties or considerable delays in system sensors, actuators, or rewards; the necessity for real-time inference at the system's control frequency; the demand for actions and policies that are explainable by system operators; reward functions that are sensitive to risks, multi-objective, or unspecified; tasks that are non-stationary or stochastic; high-dimensional continuous action and state spaces; learning on real systems with limited samples; and offline training from fixed logs of an external behavior policy.

The two metrics proposed in this study address several critical aspects of the challenges. Firstly, the Hazard Indicator provides a measure of the overall hazard associated with each state, enabling the agent to recognize and avoid states with high probabilities of leading to failure. This addresses the challenge of integrating safety considerations into the learning process and ensures that the agent does not solely focus on maximizing rewards at the expense of safety. The Risk Indicator, on the other hand, quantifies the level of risk by incorporating the cost implications of potential failure states. This helps manage risk-sensitive, multi-objective reward functions and ensures that the agent accounts for the severity of different failure states. By balancing risk and reward, the agent can make more informed decisions, thereby enhancing the safety and reliability of RL in practical applications.

By leveraging these metrics, RL agents can substantially enhance the realization and implementation of RL in practical applications, bolstering their reliability and efficacy

The rest of the paper is organized as follows: Section 2 provides an essential overview of RL and the safety aspects associated with it, addressed to readers who may encounter the concept for the first time; Section 3 briefly overviews the related work. Sections 4 and 5 present the proposed definitions and experiments, respectively. In Section 6, an application of the metric to a real scenario is given. In the last section, we draw some conclusions.

2. Technical Background

Reinforcement Learning (RL) is a computational learning approach that deals with learning in sequential decision-making problems conditioned by feedback as demonstrated in the Figure 1. It is one of three main machine learning paradigms, with supervised learning (which needs labeled input/output pairs to infer a mapping between them) and unsupervised learning (which analyzes a large quantity of unlabeled data to capture patterns from them).

RL algorithms are characterized by four basic elements: an agent, a program to train for a certain purpose; the environment, the world in which the agent can stay and performs actions, in practice, a collection of state signals; an action, a movement made by the agent that allows to switch from one state to another in the environment; and a reward signal (or simply reward), the evaluation of an action, a numerical feedback from the environment. The agent must discover which actions yield the most reward by trying them: the data are generated with a trial-and-error search during training within the environment and marked with a label. The mapping from states to actions (to be taken when in those states) is called a policy. The policy can be described as one mathematical function, a lookup table, or a more complex computational process (e.g., a neural net whose input is a representation of the state and its output an action selection probability). In other words, solving an RL task means finding a policy called optimal policy that maximizes the total reward signal over time.

Typically, an RL problem is mathematically formulated through the Markov Decision Process framework; this can be represented by a tuple (S, A, P, R, γ), where

- S is the set of all possible states.
- A is the set of all possible actions.
- P is the state transition probability function, $P(s_{t+1}|s_t, a_t)$, which represents the probability of transitioning to state s_{t+1} from state s_t by taking action a_t.
- R is the reward function, also called cost function, $R(s_t, a_t, s_{t+1})$, which provides the immediate reward received after transitioning from state s_t to state s_{t+1} due to action a_t.
- γ is the discount factor, $0 < \gamma \leq 1$, which represents the importance of future rewards.

The policy $\pi(a_t|s_t)$ defines the probability of taking action a_t when in state s_t.

The goal of the agent is to learn a policy $\pi(a_t|s_t)$ that maximize the expected discounted cumulative reward (i.e., the sum of all rewards received, scaled by a certain discount factor); in mathematical notation,

$$G = \sum_{t=0}^{T} \gamma^t \cdot R_{t+1}$$

where R_{t+1} represents the reward obtained at time step $t+1$. The initial time step is referred to as $t = 0$. The terminal step, denoted by T, can be either a finite time horizon or infinity.

Generally, to quantify the worth of a state, value functions and an action–value function are used. The value function $V(s)$ represents the expected return starting from state s, under policy π:

$$V(s) = \mathbb{E}_\pi[G_t \mid s_t = s]$$

where G_t is the return starting from time step t, and $\mathbb{E}_\pi$ denotes the expected value under policy π. The action–value function, the Q-function $Q(s,a)$, represents the expected return starting from state s, taking action a, and then following policy π:

$$Q(s,a) = \mathbb{E}_\pi[G_t \mid s_t = s, a_t = a]$$

This type of learning allows one to automate the resolution of complex control and interaction problems with the environment, in many cases not solvable with explicitly programming, potentially proposing the best possible solution.

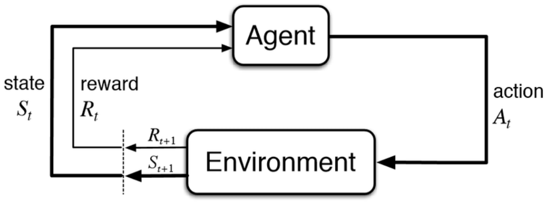


Figure 1. The reinforcement learning framework [4].

RL, while effective in complex applications, is not inherently suited for safety-critical contexts due to challenges in ensuring policy safety and exploring unfamiliar states. Exploratory strategies often rely on occasional random actions, risking unsafe outcomes. Consequently, RL in real-world safety-critical systems is typically limited to simulation software. To address this, safe RL methods have emerged. Safe RL aims to minimize the potential for harmful actions during the learning process within RL frameworks (i.e., during the process of learning policies that maximize the expectation of the return). This is especially important in real-world applications like robotics, autonomous driving, and healthcare, where unsafe actions can have serious consequences. Currently these methods can be broadly categorized into two approaches: safe optimization (also referred to as modification of the optimality criterion) and safe exploration (otherwise known as modification of the exploration process) [14].

Safe optimization integrates safety considerations into the policy, often via constraints ensuring policy parameters remain within a defined safe space. These algorithms can incorporate a risk measure of encountering unsafe situations, depending on the specific tasks. Safe exploration aims to prevent the selection of unsafe actions during training, leveraging external knowledge to guide exploration.

3. Related Work

This section briefly reviews a few research works that consider any aspect of safe reinforcement learning. A fundamental concept was given in [15], where authors introduced the idea of a probabilistic shield to improve the safety of RL algorithms. They utilized formal verification methods to calculate the probabilities of critical decisions within a safety-related fragment of Markov Decision Processes (MDPs). This work demonstrated enhanced learning efficiency in several scenarios such as service robots and the arcade game PAC-MAN. On the other hand, the authors of [16] considered the synergy between RL and model checking for scalable mission planning including many autonomous agents. Although that work was useful for complex mission planning problems, an experimental validation was missing.

A safe deep RL approach was developed in [2] in a networked microgrid system for the hierarchical coordination between individual microgrids towards decentralized operation. In the first stage of this two-stage methodology, energy storage systems and active power outputs of micro-turbines were scheduled on an hourly basis for cost minimization and energy balancing, while in the second stage, the reactive power outputs of PV inverters were dispatched every three minutes based on a Q/V droop controller to regulate voltage and to minimize network power losses under real-time uncertainties. A similar work was conducted in [17] where the authors proposed a two-stage multi-agent deep RL scheme for the reconfiguration of an urban distribution network by presenting a switch contribution concept. The authors in [18] introduced a federated learning method to integrate a transformer model and proximal algorithm for vehicle-to-grid scheduling. The authors used an end-to-end deep learning framework using current and historical data to decide scheduling under uncertainties.

Model-Agnostic Meta-Learning is employed in [19] by considering learning about tasks within a distribution, and it can adapt quickly to solve a new unseen in-distribution task. A bootstrap deep Q-network is used in [20] to learn an ensemble of Q-networks. The model also utilizes Thompson's sampling to drive exploration and enhance sample efficiency. The use of expert demonstrations to bootstrap the agent can improve sample efficiency. This methodology is applied in [21] by combining it with a deep Q-network. This method is also demonstrated on Atari games in [22]. A real system does not have a separate environment for training and evaluation, which is unlike most of the research conducted in deep RL [22]. The data for training come from the real system where an agent cannot have a separate exploration policy during training, as its exploratory actions do not come for free. In fact, an agent's performance must be reasonably good and should act safely throughout learning. This means that exploration must be limited for many systems and consequently, the resulting data have low variance—very little of the state space may be covered in the logs. Moreover, there is often only one instance of the system; schemes that instantiate hundreds or thousands of environments to collect more data for distributed training are normally not compatible with this setup [23,24]. In [25], the authors consider mobile agent path planning in uncertain environments by integrating a probabilistic model with RL.

This work proposes the Hazard Indicator and Risk Indicator as novel metrics to assess the safety of an environment. These metrics leverage the concept of transition probabilities to failure states over a specified time horizon. The Hazard Indicator provides a general measure of the likelihood of encountering any failure state from a given state, while the Risk Indicator incorporates the potential cost associated with each failure state.

The development of robust metrics for evaluating safety in RL environments is crucial for the successful deployment of RL algorithms in real-world applications.

4. Defining the Metrics

This section formally defines the metrics used to evaluate the safety of an environment with respect to failure states. We begin by defining the Markovian state space, comprising both safe and failure states, and then introduce probabilistic measures capturing the likelihood of transitioning between these states over a specified time horizon. Leveraging these probabilistic formulations, we introduce two indicators: the Hazard Indicator and the Risk Indicator. These indicators afford insights into the degree of hazard and risk associated with individual states within the system, thereby facilitating informed decision-making and risk mitigation strategies.

Safe states represent configurations within the environment where an agent's actions, or simply its presence, do not lead to any negative and undesirable consequences. These states are considered non-critical because the agent can execute automated strategies and methods without any risk.

Failure states, conversely, are configurations within the environment that pose a significant threat to the agent's structural integrity or the surrounding environment. If the agent enters a failure state, its execution of automated strategies and methods may cause damage to itself or its environment.

State space: we assume there are N possible discrete states within the environment, $S = \{s_1, \ldots, s_K, f_1, \ldots f_M\}$, $S^* = \{s_1, \ldots, s_k\}$ K being safe states, $F = \{f_1, \ldots f_M\}$ M failure states, and $N = K + M$.

Definition 1 (Transition Probability to Failure State). *The factor* $\mathrm{h}^l(f_k|s_i)$ *represents the total probability starting from state* $s_i \in S$*, at time* t*, to reach the specific failure state* $f_k \in F$*, after executing at maximum of* l *steps:*

$$\mathrm{h}^l(f_k|s_i) = \mathrm{h}^l(f_k, s_i) \doteq Prob\{S_{t+l} = f_k \in F | S_t = s_i \in S\} \tag{1}$$

In the base case, when $l = 0$, we have:

$$\mathrm{h}^l(f_k|s_i) = \begin{cases} 1 & s_i \in F \\ 0 & s_i \in S^* \end{cases}$$

Definition 2 (Hazard Indicator). *The Hazard Indicator, denoted by* H_i^l*, represents the overall degree of hazard associated with that state* s_i*:*

$$H_i^l = \sum_{k=1}^{M} \mathrm{h}^j(f_k, s_i) \tag{2}$$

This indicator is particularly informative when all failure states M carry equal cost. If this assumption does not hold, it becomes crucial to consider the associated costs. Let us assume $C = \{c_1, \ldots c_M\}$ are the costs associated with the M failure states, respectively.

Definition 3 (Risk Indicator)**.** *The Risk Indicator, denoted by R_i^l, quantifies the level of risk associated with the state s_i, by integrating the cost implications associated with individual potential failure states:*

$$R_i^l = \sum_{k=1}^{M} c_k \, \mathrm{h}^j(f_k, s_i) \tag{3}$$

Note: It is essential to acknowledge that the value of the parameter l is contextually dependent. However, the experimental phase revealed that excessively large values of l could compromise the effectiveness of the indicator. Accordingly, it is advisable to select values of l that enable us to operate in close proximity to failure states. This optimization ensures the metric's practical usefulness.

5. Experiments

In the experimental phase, we considered the interaction of an agent within a simulation environment modeled as a stationary Markov Decision Process (MDP). This study employed a mobile robot model navigating a spatial environment discretized into states represented by a grid structure. Various scenario configurations were utilized to evaluate the efficacy and adaptability of the aforementioned Hazard and Risk Indicators.

Case 1 The agent navigates a grid with a single failing terminal state. All possible actions sampled by the agent are equiprobable, meaning each action has a 25% chance of being selected.

Case 2 Similar to Case 1, all actions are equiprobable; however, in this case, the grid contains two failing terminal states with identical associated costs.

Case 3 The grid structure remains the same as Case 2, featuring two failing terminal states. However, the costs associated with reaching each terminal state differ.

Case 4 The grid configuration reverts to a single failing terminal state. However, we introduced some systemic disturbing factors: for each step, the agent has a 70% probability of moving up. There is also a 10% chance of transitioning to any of the other three directions (down, left, or right).

5.1. Design of Experiments

The overarching goal of these experiments was to evaluate the effectiveness of the Hazard Indicator and Risk Indicator within domains that can be effectively represented using a Markovian framework. To achieve this, a two-dimensional grid-world environment was constructed. This world consisted of a 10 × 6 grid, yielding a total of 60 accessible states. Depending on the experimental condition, one or two of these states were designated as failure states, while the remaining states were considered safe. The agent's action space was comprised of four discrete actions: up, down, right, and left. In each experiment, the following conditions held:

- Limited exploration steps: the agent was restricted to a maximum of three actions, or more precisely three state transitions, i.e., $l = 3$.
- Boundary handling: upward movements directed towards the grid's edges resulted in the agent's position remaining unchanged.
- Early termination upon failure encounter: if the agent encountered a failure state during the first step, exploration was immediately terminated.

5.2. Experiment: Case 1

To promote a comprehensive understanding of the proposed approach, we commenced our investigation with a well-defined scenario. Within this scenario, only one of the 60 accessible states was designated as a failure state. Furthermore, the probability distribution governing the agent's actions was uniform across all states. In simpler terms, each action (up, down, left, right) had an equiprobable chance of being selected. In the heatmap shown in Figure 2, the Hazard Indicator values obtained, $H_i^3 \;\; \forall i$, are displayed. These values represent the hazard level associated with each state.

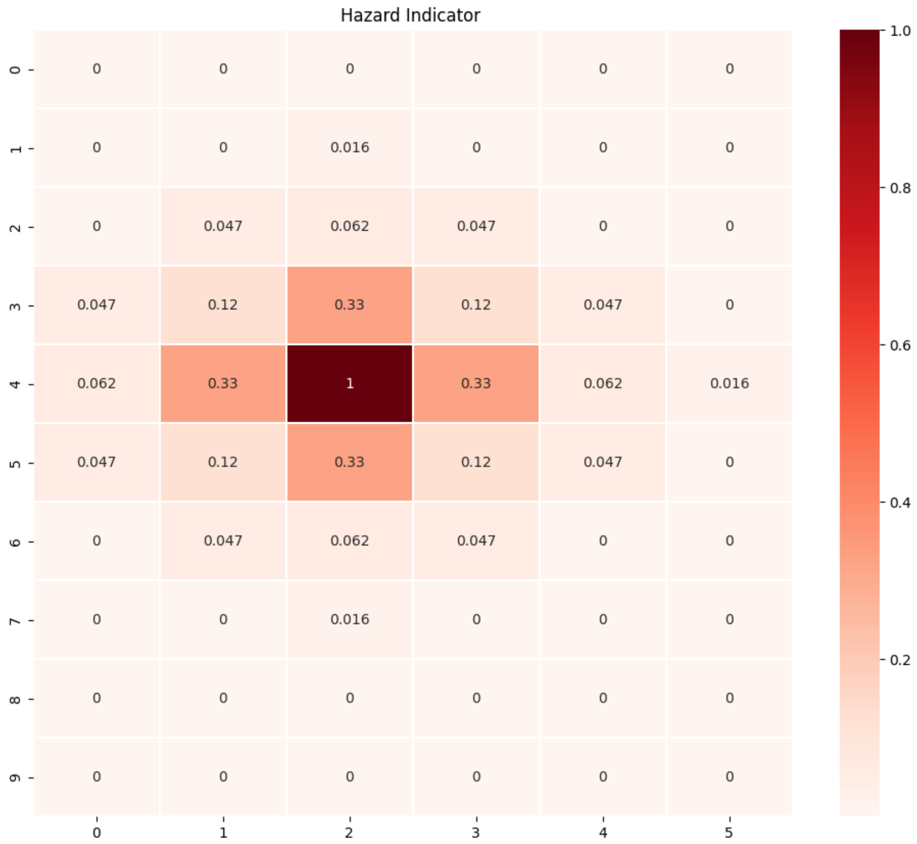


Figure 2. It is noteworthy that the safest zones are denoted with values of 0; while the failure state, row 4 and column 2, holds a value of 1, as per its definition. The remaining states show a different degree of risk. Interestingly, the index values reveal a bell-shaped centered on the failure state.

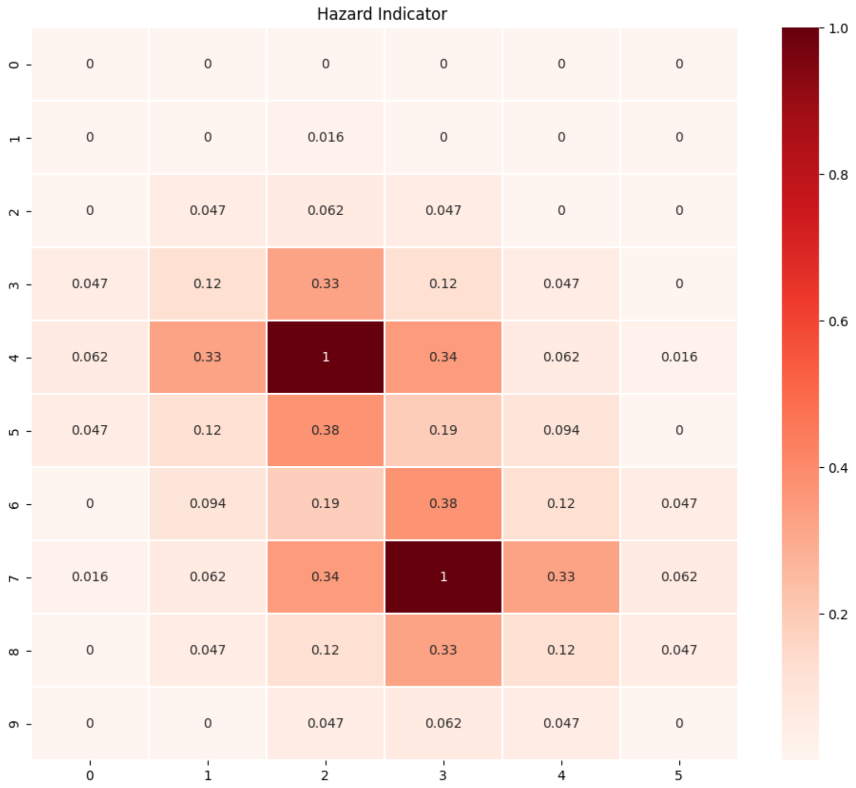
To illustrate the results, consider the state located at row 5 and column 2. From this state, there exist six paths with a maximum of three steps that lead to the failure state. These paths are enumerated as follows: [′right′, ′left′, ′up′], [′right′, ′up′, ′left′], [′left′, ′right′, ′up′], [′left′, ′up′, ′right′], [′up′], and [′down′, ′up′, ′up′]. The total probability of reaching the failure state from this starting state can be calculated as follows: (0.016 × 5) + (0.25). The first term represents the sum of probabilities for the five paths (excluding [′up′]). Each of these paths has a probability approximately equal to 0.016 (obtained by multiplying the individual action probabilities: 0.25 × 0.25 × 0.25). The second term (0.25) represents the probability of reaching the failure state directly in a single step: ′up′.

5.3. Experiment: Case 2

To introduce a level of complexity, we opted to work with two failure states. These failure states were assigned equivalent costs. The heatmap in Figure 3 presents the obtained results from this scenario.

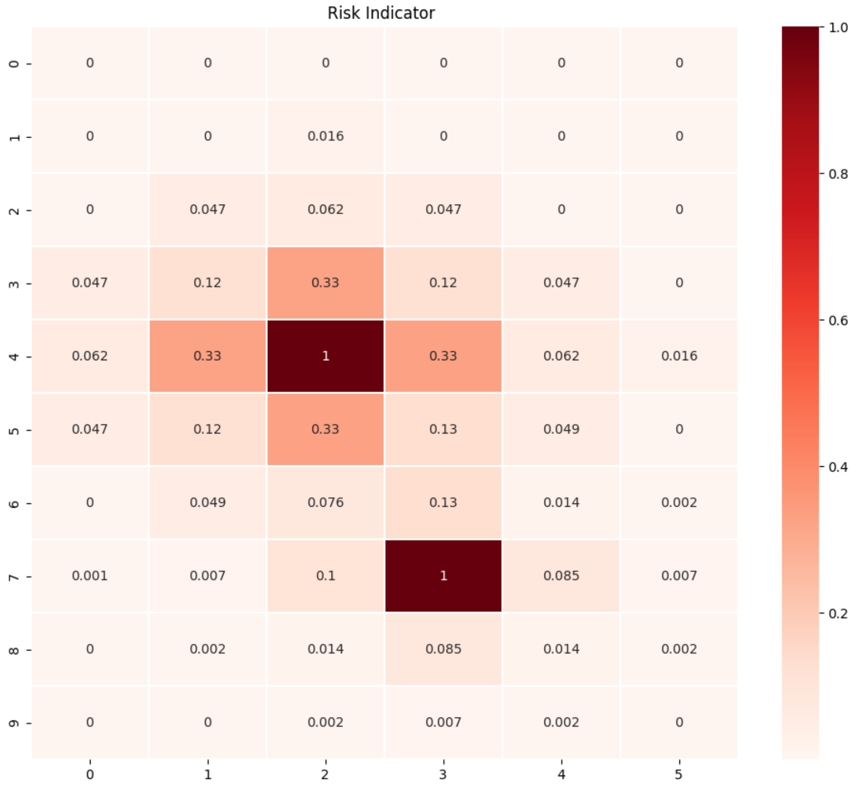
5.4. Experiment: Case 3

Next, we assigned distinct costs to the two failure states. Specifically, one failure state was designated as having a cost one-third lower than the other. In that scenario, the Risk Indicator was employed. The Figure 4 presents the results obtained.



	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0.016	0	0	0
2	0	0.047	0.062	0.047	0	0
3	0.047	0.12	0.33	0.12	0.047	0
4	0.062	0.33	1	0.34	0.062	0.016
5	0.047	0.12	0.38	0.19	0.094	0
6	0	0.094	0.19	0.38	0.12	0.047
7	0.016	0.062	0.34	1	0.33	0.062
8	0	0.047	0.12	0.33	0.12	0.047
9	0	0	0.047	0.062	0.047	0

Figure 3. The figure shows a scenario with two failure states. The states included among these are characterized by the highest hazard indices.



	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0.016	0	0	0
2	0	0.047	0.062	0.047	0	0
3	0.047	0.12	0.33	0.12	0.047	0
4	0.062	0.33	1	0.33	0.062	0.016
5	0.047	0.12	0.33	0.13	0.049	0
6	0	0.049	0.076	0.13	0.014	0.002
7	0.001	0.007	0.1	1	0.085	0.007
8	0	0.002	0.014	0.085	0.014	0.002
9	0	0	0.002	0.007	0.002	0

Figure 4. In this grid, there are two failure states with distinct costs: the state located at row 7 and column 3 carries one-third of the cost of the state at row 4 and column 2. Notably, we observe that higher risk values dominate the regions around this latter state.

5.5. Experiment: Case 4

In this experimental session, we altered the probability distribution of actions with a systemic disturbance factor, moving away from a uniform distribution. Specifically, the agent had a 70% probability of moving upward, a 10% probability of moving downward, a 10% probability of moving left, and a 10% probability of moving right. As in the previously examined scenario, we considered one failure state, as presented in Figure 5.

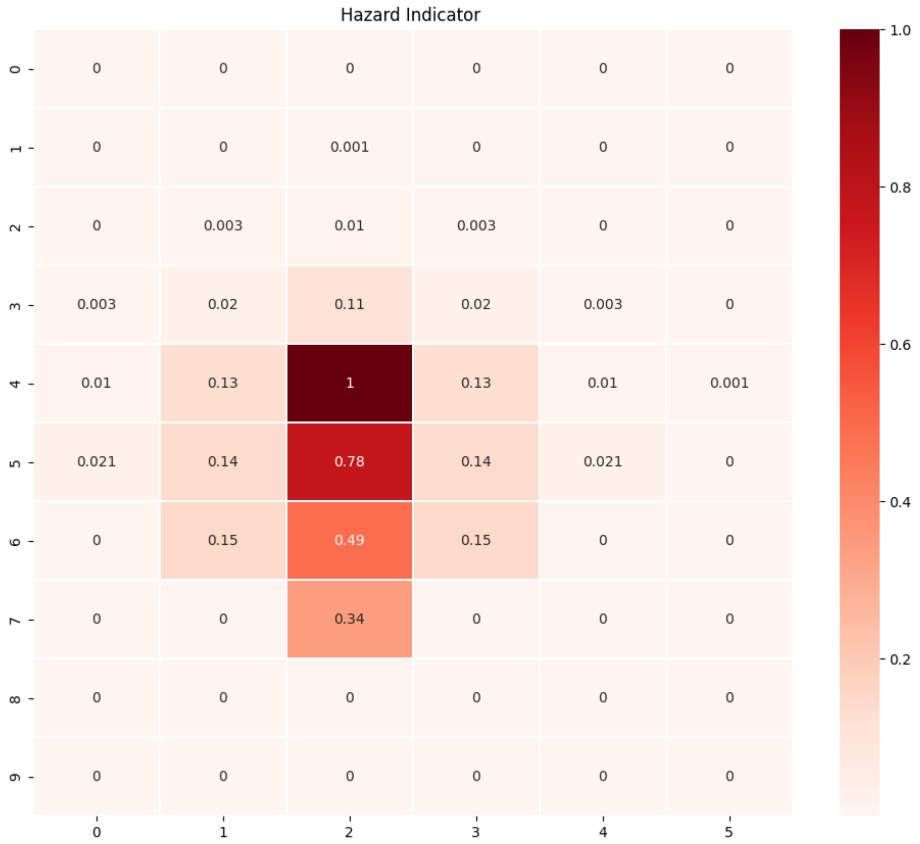


Figure 5. In this grid, we observe that the state surrounding the failure states, associated with the "up" direction, exhibits a higher index value.

5.6. Assessment of Experimental Outcomes

We utilized the values of these indicators by experimenting with simulations. Our objective was for the agent, starting from the initial state of the grid at [0, 0], to reach the target state at [9, 4] on the opposite side of the grid, near a failing state, in the minimum number of steps while avoiding fatal situations. We implemented two of the most widely used RL algorithms, with different operational principles: on-policy first-visit Monte Carlo control, and Q-learning one-step (off-policy temporal-difference control), both implemented in look-up-table form and using an epsilon-greedy policy. More in-depth details of these methods can be found in [4].

During the training phase, the sampling of each action considered the values of the Hazard or Risk Indicators, depending on the scenario. When these values exceeded a predefined threshold (e.g., in experimental case 1, the threshold was set to either 0.12 or 0.33 depending on the desired proximity to potential failure states), the sampling switched from the epsilon-greedy exploration strategy (basically, a uniform random exploration in the early steps of training) to a guided mode to avoid visiting high-risk states, remaining in states with the index indicative of lower risk, and effectively creating a path that ensured system safety. This approach significantly influenced the agent's behavior: by adjusting the threshold levels, it reduced or completely eliminated the possibility of reaching a failure state. Concurrently, an optimal policy for the target was obtained, ensuring the achievement of the desired state in the minimum number of steps while prioritizing safety.

6. Use Case

This section investigates the application of the Hazard Indicator by examining its effectiveness in a real-world scenario: risk management during medical examinations at a Nuclear Medicine Department (NMD). It refers to the practices and protocol implemented within a Nuclear Medicine Department to minimize potential risks associated with the administration of radio-pharmaceuticals for diagnostic procedures. The inherent risk of radiation exposure necessitates close patient monitoring and guidance by nurses during their movements to prevent hazardous situations. For a comprehensive understanding, it is important to acknowledge that the NMD encompasses several distinct locations:

- The reception room (RR), where patients are admitted into the department.
- The waiting room (WR), designated for patients to wait before the injection of a radio-pharmaceutical.
- The injection room (IR), where patients receive the necessary substance.
- The hot waiting room (HWR), where patients wait until their radiation levels reach the target range post-injection.
- The diagnostic room (DR), where examinations are conducted.

For the medical procedure to be successfully completed, patients must follow this specific sequence: RR → WR → IR → HWR → DR. If a patient deviates from the prescribed sequence, they take a risk of exposure to radioactive agents, which can cause severe injury. Traditionally, nurses ensure adherence to the established procedural sequence during examinations. However, automating this process represents a paradigm shift, transitioning away from the dependence on continuous patient monitoring by hospital staff. To achieve this, various adaptive decision-making controllers, acting as intelligent agents empowered by reinforcement learning algorithms, have demonstrated significant efficacy [26,27].

To evaluate the effectiveness of the proposed metric, we employed the system outlined in Shah's work [27]. Specifically, the controller utilized RL methods to adapt to the dynamic environment of the examination building. The controller received information about the patient's current location and based on its learned understanding of the environment subsequently sent guidance messages directly to the patient's smartphone. A comprehensive description of the employed Markov Decision Process framework is provided in Figure 6, which illustrates the state–action architecture, and in Table 1, which provides a comprehensive description of the assumed scenario.

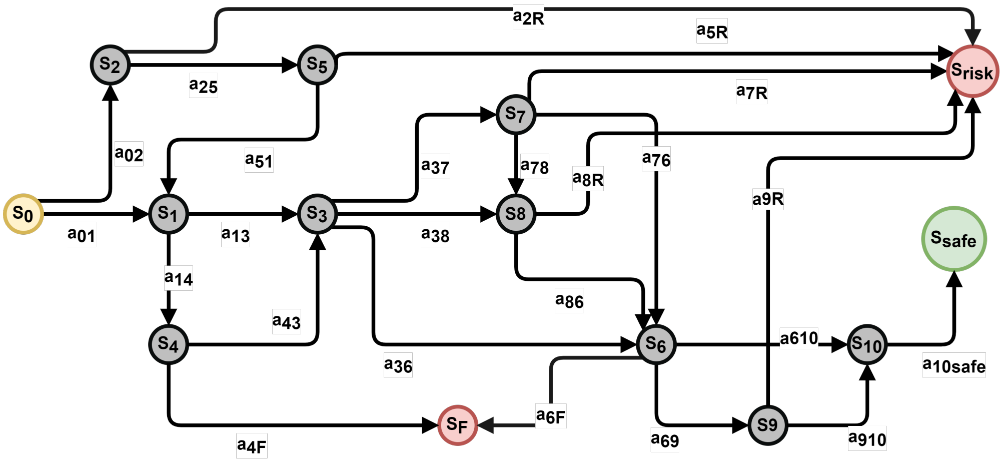


Figure 6. Risk case scenario: this figure (graph-based) illustrates a model's state–action representation for risk management within a Nuclear Medicine Department.

Table 1. Description of states and actions in the Nuclear Medicine Department scenario.

States	Description	Actions	Description
s_0	Patient admitted for medical examination	a_{01}	You are in the acceptance room; please move to the waiting room
		a_{02}	You are in the acceptance room; please move to the hot waiting room
s_1	A patient waits in the WR and is not injected		You are in the waiting room; please move to the injection room
		a_{14}	You are moving to the same waiting room
s_2	Patient waits in the HWR and is not injected	a_{25}	Be careful!, You are moving to the HWR without being injected
		a_{2R}	You are in a risk state
s_3	Patient is available for injection in the IR	a_{36}	You are in the injection room; please move to the hot waiting room
		a_{37}	You are moving back to the waiting room; please move to the HWR
		a_{38}	Your are being injected and in the WR, please move to the HWR
s_4	A patient exits the WR but enters the WR again	a_{4F}	The system does not work
s_5	The patient goes in and out of the HWR without being injected	a_{51}	You are in the HWR without being injected; please move to the WR
		a_{5R}	You are in a risk state
s_6	The patient waits in the HWR and is being injected	a_{6F}	The system does not work
		a_{69}	You are still in the HWR; please move to the diagnostic room
		a_{610}	You are in the HWR; please move to the diagnostic room
s_7	The patient waits in the WR and is being injected	a_{7R}	You are in a risk state
		a_{76}	Your are being injected and in the WR; please move to the HWR
		a_{78}	Be careful! Your are being injected and in the WR, please move to the HWR
s_8	The patient goes in and out of the WR after being injected	a_{8R}	You are in a risk state
		a_{86}	Your have been injected and are in the WR, please move to the HWR
s_9	The injected patient waits in the HWR instead of moving into the DR	a_{9R}	You are in a risk state
		a_{910}	Please move to the diagnostic room for inspection
s_{10}	Patient under examination	a_{10Safe}	You are in the diagnostic room; the examination process is underway
		Terminal States	
s_{safe}	A patient exits from the DR		
s_F	System failure		
s_R	Risk position		

In essence, the environment was modeled with fourteen distinct states, two of which represented terminal states that posed a significant risk to the patient's safety. Due to these inherent dangers associated with the task and the relatively small number of states, we opted to commence our investigation with an agent restricted to a single action-step, i.e., $l = 1$. Additionally, each action had the same probability of being sampled from those available.

The graph in Figure 7 depicts the results obtained by employing the Hazard Indicator metric, $l = 1$, within the scenario described.

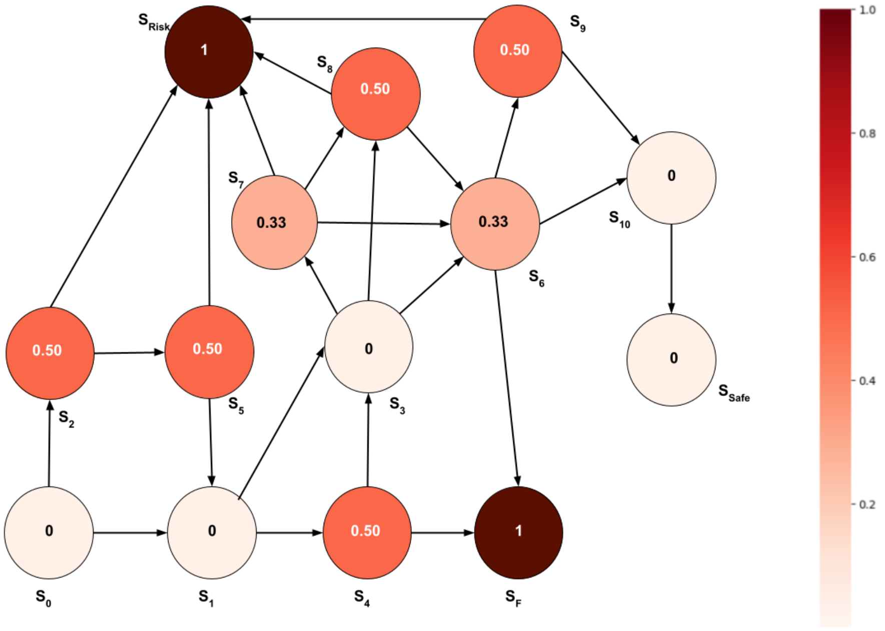


Figure 7. Hazard Indicator values with $l = 1$. With reference to the state–action architecture detailed above, the figure illustrates the corresponding Hazard Indicator for each state.

The graph in Figure 7 employs a visual representation where failure states are depicted with a value of one. Conversely, a value of zero indicates safe states. The remaining states are assigned values between zero and one, reflecting varying degrees of hazard.

To further investigate the capabilities of the Hazard Indicator, we expanded the agent's decision-making horizon by increasing the maximum number of allowable actions from one to two steps, i.e., $l = 2$. The results are shown in Figure 8.

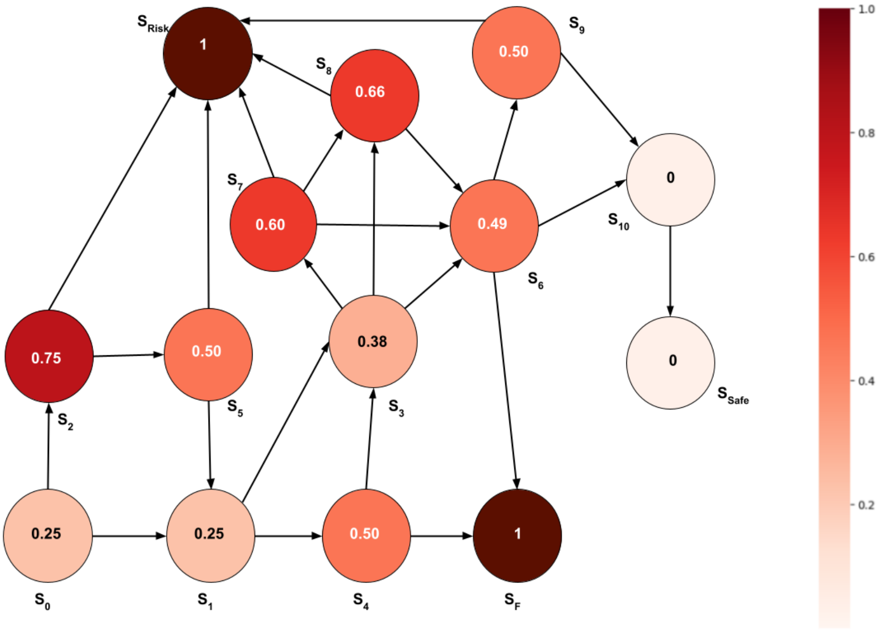


Figure 8. Hazard Indicator values with $l = 2$.

In this scenario, the graph revealed a more nuanced articulation of the Hazard Indicator, capturing a wider spectrum of risk levels compared to the previous scenario.

For our experiments, we constructed the simulated Markovian environment described previously. We employed the same on-policy control algorithm, State–Action–Reward–State–Action (SARSA), utilizing an epsilon-greedy policy implemented via a lookup table, as used in [27]. A comprehensive explanation of these RL techniques can be found in [4]. However, we differentiated our approach by integrating the SARSA algorithm with the Hazard Indicator values. We implemented a threshold mechanism that triggered a piloted exploration when the index exceeded a predefined value. By doing so, it ensured the agent remained in states where the Hazard Indicator values confirmed a higher level of safety. Having exploited the SARSA algorithm, we focused on the scenario where $l = 1$. By modulating this threshold, we aimed to manipulate the agent's exposure to potentially risky situations during training. We investigated the impact of two threshold values: 0.33 and 0.50. Only the first one guaranteed a total safe training by preventing the agent from encountering potentially hazardous scenarios. The second threshold, 0.5, reduced the frequency of encountering a failure state during training, compared to an algorithm that did not use the index. This approach enabled the convergence to the same optimal policy achieved by the original implementation of the algorithm that did not use safety precautions.

7. Conclusions

This paper introduced a novel metric designed to assess the risk within environments characterized by Markov Decision Processes.

The Hazard Indicator and Risk Indicator proposed in this research offer quantitative evaluations of the risk associated with individual states. These indicators enable agents to prioritize safer actions and interventions based on index values.

While established techniques such as safe exploration methods like shielding rely on predefined safety models, these indicators offer greater flexibility and adaptability across different environments without the need for specific safety rules. Reward shaping and constrained optimization, while effective, often necessitate customization for individual tasks. In contrast, Hazard and Risk Indicators demonstrate greater versatility across a wider range of RL applications. Furthermore, some safe exploration techniques might require complex model training, while these indicators provide a simpler approach to quantifying and managing risk. Therefore, ultimately, their straightforward implementation, highly generalized probabilistic mathematical formulation, and task-independent nature facilitate the RL agent's comprehension of hazardous states within an environment.

The experiments conducted validated the efficacy of these metrics across various scenarios, elucidating their utility in assessing risk and hazard in complex environments, thereby providing valuable insights for decision-making and risk mitigation strategies across diverse domains.

The primary limitation of these metrics stems from their underlying assumption: a discrete and fully observable state space. In complex environments with continuous or partially observable states, the applicability of these metrics becomes limited. This also poses computational challenges as the number of states increases.

In our future research, we will be actively exploring methods to address these limitations. This includes researching the use of function approximation techniques to refine these metrics and incorporating probabilistic methods to handle partially observable environments. Additionally, exploring the integration of these indicators with a variety of algorithms, beyond those associated with the RL framework, presents a promising avenue for future research.

Author Contributions: Conceptualization, A.C. and M.C.; methodology, M.F., A.C. and M.N.; investigation, M.F., M.N. and M.C.; writing—original draft preparation, M.F. and M.N.; writing—review and editing, M.N. and A.C.; supervision, M.C. and A.C.; funding acquisition, M.C. and A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially supported by the Piano Nazionale di Ripresa e resilienza (PNRR)—Missione 4 "Istruzione e Ricerca"—Componente C2—Investimento 1.1, "Fondo per il Programma Nazionale di Ricerca e Progetti di Rilevante Interesse Nazionale (PRIN)"—Grant number P2022MXCJ2, Settore PE1—LINEA B—Sud, CUP F53D23010080001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Naeem, M.; Coronato, A.; Paragliola, G. Adaptive treatment assisting system for patients using machine learning. In Proceedings of the 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, 22–25 October 2019; pp. 460–465.
2. Xia, Y.; Xu, Y.; Feng, X. Hierarchical Coordination of Networked-Microgrids towards Decentralized Operation: A Safe Deep Reinforcement Learning Method. *IEEE Trans. Sustain. Energy* **2024**, *15*, 1981–1993. [CrossRef]
3. Xia, Y.; Xu, Y.; Wang, Y.; Mondal, S.; Dasgupta, S.; Gupta, A.K.; Gupta, G.M. A safe policy learning-based method for decentralized and economic frequency control in isolated networked-microgrid systems. *IEEE Trans. Sustain. Energy* **2022**, *13*, 1982–1993. [CrossRef]
4. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
5. Naeem, M.; Rizvi, S.T.H.; Coronato, A. A gentle introduction to reinforcement learning and its application in different fields. *IEEE Access* **2020**, *8*, 209320–209344. [CrossRef]
6. Shah, S.I.H.; Coronato, A.; Naeem, M.; De Pietro, G. Learning and Assessing Optimal Dynamic Treatment Regimes Through Cooperative Imitation Learning. *IEEE Access* **2022**, *10*, 78148–78158. [CrossRef]
7. Jamal, M.; Ullah, Z.; Naeem, M.; Abbas, M.; Coronato, A. A Hybrid Multi-Agent Reinforcement Learning Approach for Spectrum Sharing in Vehicular Networks. *Future Interent* **2024**, *16*, 152. [CrossRef]

8. Amin, M.S.; Rizvi, S.T.H. Sign gesture classification and recognition using machine learning. *Cybern. Syst.* **2023**, *54*, 604–618. [CrossRef]
9. Naeem, M.; Coronato, A. An AI-empowered home-infrastructure to minimize medication errors. *J. Sens. Actuator Netw.* **2022**, *11*, 13. [CrossRef]
10. Naeem, M.; Coronato, A.; Ullah, Z.; Bashir, S.; Paragliola, G. Optimal User Scheduling in Multi Antenna System Using Multi Agent Reinforcement Learning. *Sensors* **2022**, *22*, 8278. [CrossRef] [PubMed]
11. Gavade, A.B.; Nerli, R.; Kanwal, N.; Gavade, P.A.; Pol, S.S.; Rizvi, S.T.H. Automated diagnosis of prostate cancer using mpmri images: A deep learning approach for clinical decision support. *Computers* **2023**, *12*, 152. [CrossRef]
12. Coronato, A.; Naeem, M. A reinforcement learning based intelligent system for the healthcare treatment assistance of patients with disabilities. In *International Symposium on Pervasive Systems, Algorithms and Networks*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 15–28.
13. Kwon, R.; Kwon, G.; Park, S.; Chang, J.; Jo, S. Applying Quantitative Model Checking to Analyze Safety in Reinforcement Learning. *IEEE Access* **2024**, *12*, 18957–18971. [CrossRef]
14. Garcıa, J.; Fernández, F. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.* **2015**, *16*, 1437–1480.
15. Jansen, N.; Könighofer, B.; Junges, S.; Serban, A.; Bloem, R. Safe reinforcement learning using probabilistic shields. In Proceedings of the 31st International Conference on Concurrency Theory (CONCUR 2020), Schloss-Dagstuhl-Leibniz Zentrum für Informatik, Online, 1-4 September 2020.
16. Gu, R.; Enoiu, E.P.; Seceleanu, C.; Lundqvist, K. *Combining Model Checking and Reinforcement Learning for Scalable Mission Planning of Autonomous Agents*; Mälardalen Real-Time Research Centre, Mälardalen University: Västerås, Sweden, 2020; Available online: https://www.es.mdu.se/publications/5782-Combining_Model_Checking_and_Reinforcement_Learning_for_Scalable_Mission_Planning_of_Autonomous_Agents (accessed on 15 January 2024).
17. Gao, H.; Jiang, S.; Li, Z.; Wang, R.; Liu, Y.; Liu, J. A Two-stage Multi-agent Deep Reinforcement Learning Method for Urban Distribution Network Reconfiguration Considering Switch Contribution. *IEEE Trans. Power Syst.* **2024**, 1–12. [CrossRef]
18. Shang, Y.; Li, S. FedPT-V2G: Security enhanced federated transformer learning for real-time V2G dispatch with non-IID data. *Appl. Energy* **2024**, *358*, 122626. [CrossRef]
19. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
20. Osband, I.; Blundell, C.; Pritzel, A.; Van Roy, B. Deep exploration via bootstrapped DQN. *Adv. Neural Inf. Process. Syst.* **2016**, *29* .
21. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
22. Hester, T.; Vecerik, M.; Pietquin, O.; Lanctot, M.; Schaul, T.; Piot, B.; Horgan, D.; Quan, J.; Sendonaris, A.; Osband, I.; et al. Deep q-learning from demonstrations. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
23. Horgan, D.; Quan, J.; Budden, D.; Barth-Maron, G.; Hessel, M.; Van Hasselt, H.; Silver, D. Distributed prioritized experience replay. *arXiv* **2018**, arXiv:1803.00933.
24. Espeholt, L.; Soyer, H.; Munos, R.; Simonyan, K.; Mnih, V.; Ward, T.; Doron, Y.; Firoiu, V.; Harley, T.; Dunning, I.; et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1407–1416.
25. Wang, X.; Liu, J.; Nugent, C.; Cleland, I.; Xu, Y. Mobile agent path planning under uncertain environment using reinforcement learning and probabilistic model checking. *Knowl.-Based Syst.* **2023**, *264*, 110355. [CrossRef]
26. Paragliola, G.; Coronato, A.; Naeem, M.; De Pietro, G. A reinforcement learning-based approach for the risk management of e-health environments: A case study. In Proceedings of the 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Las Palmas de Gran Canaria, Spain, 26–29 November 2018; pp. 711–716.
27. Shah, S.I.H.; Naeem, M.; Paragliola, G.; Coronato, A.; Pechenizkiy, M. An AI-empowered infrastructure for risk prevention during medical examination. *Expert Syst. Appl.* **2023**, *225*, 120048. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.