

Resilient Multi-Agent Systems for Environmental Monitoring

Egidia Cirillo^{1,*}, Alessandra Amato¹, Alberto Moccardi¹ and Marcello Pelosi¹

¹Department of Electrical Engineering and Information Technology (DIETI), University of Naples Federico II, Via Claudio 21, 80125 Naples, Italy

Abstract

Climate variability and the increasing frequency of environmental shocks highlight the fragility of conventional agricultural and environmental monitoring systems, which typically depend on static thresholds and single-point decision logic. A Resilient Multi-Agent System (R-MAS) architecture is proposed, combining classical Belief–Desire–Intention agents, predictive components powered by Agentic AI, and a human-in-the-loop oversight layer to enable continuous adaptation under uncertainty. The architecture is structured around a four-phase *Resilience Loop* (Monitor–Detect–Adapt–Learn), designed to identify drift, plan recovery actions, and update agent policies after each shock. A trust and explainability mechanism is integrated to generate post-hoc, agent-level justifications for actions and to support human override. The system is evaluated in a simulated smart-agriculture scenario exposed to four categories of shock, drought, heatwave, sensor failure, and supply disruption and compared with both a static rule-based baseline and a purely reactive multi-agent baseline. Results indicate that the R-MAS reduces recovery time by a factor of 2–3.5, maintains water-use efficiency above 0.65 during a 15-day drought event (compared to a drop below 0.40 for the static baseline), and, with simulated operator profiles, raises user trust scores by 35–40 percentage points when explainability and human override are enabled. In the absence of an experimental evaluation involving real users, the observed increase in trust scores cannot be regarded as a field-validated measure; rather, it should be interpreted solely as an estimate of the potential impact resulting from the integration of explainability mechanisms and human intervention capabilities. Implications for trustworthy environmental AI are discussed, along with limitations and ethical considerations related to the deployment of autonomous agents in socio-ecological contexts.

Keywords

multi-agent systems, agentic AI, resilient AI, climate adaptation, smart agriculture, trustworthy AI, explainability, human-in-the-loop

1. Introduction

Environmental and agricultural systems are facing increasing pressure due to climate change. The Sixth Assessment Report of the Intergovernmental Panel on Climate Change highlights an increase in both the frequency and intensity of extreme events, such as heatwaves, droughts, floods and supply chain disruptions, which have knock-on effects on food, water and energy [1]. At the same time, the rapid digitalisation of agriculture and environmental monitoring has led to a vast amount of real-time data from soil sensors, weather stations, satellite imagery and connected machinery [2, 3]. Despite this abundance of data, many decision-support tools still rely on fixed rules and simple models, which perform poorly when conditions change or when certain sensors fail.

In this context, Multi-Agent Systems (MAS) represent a natural way to manage distributed and complex systems. The idea is to represent each component, whether physical or logical, as an autonomous agent, capable of making decisions and communicating with others [4, 5]. In recent years, advances in language models and agentic AI have made these agents even more powerful, enabling them to reason even with incomplete information, plan actions and explain their decisions in natural language [6, 7, 8]. However, these two lines of research, taken individually, are insufficient for real-world and critical ap-

WOA 2026, the 27th Workshop From Objects to Agents, June 2026

*Corresponding author.

✉ egidia.cirillo@unina.it (E. Cirillo); alessandra.amato@unina.it (A. Amato); alberto.moccardi@unina.it (A. Moccardi); marcello.pelosi2@unina.it (M. Pelosi)

ORCID 0009-0005-3227-6073 (E. Cirillo); 0009-0001-6136-9368 (A. Moccardi); 0009-0001-4718-4826 (M. Pelosi)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

plications. Traditional MAS are inflexible in anticipating unforeseen events, whilst AI-based approaches often lack a clear structure, reliable coordination mechanisms and an explicit model of resilience.

In this work, we propose a resilient multi-agent system (R-MAS) for environmental monitoring and adaptation to climate change. The system consists of several agents with distinct roles, which collaborate to observe the environment, make decisions and adapt over time. Its operation is based on a continuous cycle that allows the system to monitor the situation, identify potential problems, react and learn from experience. Furthermore, particular attention is paid to transparency and trust: every decision is accompanied by an explanation, and the user can always intervene and modify the proposed actions.

The main contributions of this work are as follows. Firstly, we propose a multi-agent architecture designed to be resilient, in which the various tasks (reaction, prediction and decision-making) are separated and assigned to specific agents. Secondly, we introduce a resilience model that allows the system to adapt continuously to unforeseen events. Thirdly, we integrate a mechanism for human explanation and control, which makes the system more transparent and reliable. Finally, we evaluate the system in various simulated scenarios, demonstrating significant improvements in terms of efficiency, adaptation speed, stability and user trust.

The main contribution of this work is not the cyclical control structure itself—which is already widely used in multi-agent systems—but the way in which it is integrated with three key elements. The first is a quantitative, time-dependent resilience metric (Eq. 2), which allows for the objective measurement of the system’s ability to recover following a shock. The second is a learning-based adaptation phase, implemented using cooperative Multi-Agent Reinforcement Learning (MARL) techniques, which allow agents to improve their strategies over time. The third is a mechanism for explainability and human supervision at the individual agent level, the contribution of which is analyzed separately via an ablation study presented in (Section 4). It is important to note that the evaluation was conducted entirely in a simulated environment, using simulated operator profiles. Consequently, the results obtained allow us to assess the effectiveness of the proposed mechanisms and their interactions, but cannot be considered a demonstration of the system’s performance in real-world contexts. Field validation therefore represents an important direction for future work.

Overall, this work demonstrates how the combination of intelligent agents, adaptability and human supervision can lead to systems that are more robust, reliable and better equipped to tackle current environmental challenges.

2. Related Work

Multi-Agent Systems (MAS) have been used for many years to model distributed decision-making processes, particularly in environmental and agricultural contexts, where various elements must interact with one another. Early studies introduced the Belief–Desire–Intention (BDI) paradigm [9], which remains an important foundation for the construction of autonomous agents [10, 11]. In these contexts, MAS have been applied to problems such as irrigation management, pest control and supply chain coordination [2, 3]. However, many of these approaches assume relatively stable environments and focus on reactive behaviours, without explicitly considering the ability to adapt to sudden changes or system failures.

In recent years, advances in language models (LLMs) [12], transformer architecture [13] and agentic AI have enabled the development of more advanced agents, capable of reasoning with incomplete information, planning actions, utilising external tools, and explaining their decisions in natural language [6, 8]. Techniques such as retrieval-augmented generation also allow for the integration of up-to-date external data, such as weather forecasts or sensor data [14]. Despite this potential, many of these systems are tested in controlled environments and do not fully address the requirements of critical applications, where robustness, coordination and transparency are essential.

In parallel, Multi-Agent Reinforcement Learning (MARL) has been proposed as a way to enable agents to learn coordinated behaviours. These methods have been successfully applied in sectors such as traffic

and energy [15, 16]. However, systems based solely on learning may struggle in highly uncertain or dynamic environments, particularly when data is limited or unexpected events occur. For this reason, hybrid approaches combining machine learning and predefined rules are emerging.

Another important concept is that of resilience, which originated in ecological studies and refers to a system’s ability to continue functioning even in the presence of disturbances [17]. This concept has also been extended to socio-ecological and technological systems, where it is often measured by the ability to recover quickly following a shock [18, 19]. Henry and Ramirez-Marquez [20] proposed a quantitative time-dependent resilience metric for engineered systems. We adopt this formulation, rather than a steady-state or return-to-equilibrium definition, precisely because it is *time-dependent*: it captures the depth of the post-shock drop, the speed of recovery and any overshoot in a single normalized quantity, which matches our goal of comparing how quickly competing policies restore function after a shock. We instantiate its performance function Φ as water-use efficiency (yield per cubic metre of water), the quantity most directly degraded by the shocks we study; the metric is agnostic to this choice and any other scalar performance signal could be substituted. Despite its importance, resilience is still rarely considered a central element in the design of MAS.

Finally, in recent years there has been growing attention towards trustworthy AI, i.e. artificial intelligence systems that are transparent, robust and under human control [21]. Existing explainability techniques focus primarily on explaining the decisions of individual models [22, 23]. However, in multi-agent systems, decisions depend on the interaction between multiple components, making it necessary to provide explanations at the level of agent behaviour and their interactions [24].

Beyond single-model explainability, a growing body of work pursues trustworthiness at the level of the learning process itself, for example through federated and privacy-preserving paradigms [25, 26] and through mechanisms for reliability and the right to remove data such as federated unlearning [27].

Our work fits into this context, combining MAS, agentic AI and resilience within a single framework. In particular, we propose a system that integrates adaptive capabilities, explicit resilience mechanisms and explanations at the agent level, with the aim of supporting reliable operation in dynamic and complex environments [28, 29].

3. Methodology

3.1. Problem Definition

Consider a system that must manage an environment (e.g. an agricultural one) over time. The operation is divided into discrete steps, from $t = 1$ up to T . At each step, the system observes the state of the environment and must decide what to do. In particular, at every instant the system receives three types of information:

- data from sensors $\mathbf{o}_t \in \mathbb{R}^n$, such as soil moisture, temperature or CO₂ concentration;
- the climatic context $\mathbf{c}_t$, for example weather forecasts, season or geographical location;
- the state of the actuators $\mathbf{u}_t \in \mathcal{U}$, such as valve openings or ventilation settings.

Based on this information, the system must choose an action $\mathbf{a}_t \in \mathcal{A}$. This decision will influence the state of the environment at the next step. However, the behaviour of the environment is not entirely predictable and is described probabilistically by the function:

$$P(\mathbf{o}_{t+1} \mid \mathbf{o}_t, \mathbf{a}_t, \mathbf{c}_t)$$

In other words, we cannot know for certain what will happen after an action, but only estimate it. The problem becomes even more difficult because we must manage different types of uncertainty:

- Aleatoric uncertainty: due to natural variability, such as changes in the weather.
- Epistemic uncertainty: linked to incomplete knowledge of the system, for example soil or crop parameters that are not perfectly known.

- Structural shocks: sudden and rare events, such as droughts, heatwaves, sensor failures or supply chain issues.

A simple approach based on fixed rules, denoted by π_{static} , does not work well in this context: it is unable to predict shocks and struggles to adapt when something goes wrong.

A purely reactive multi-agent system improves the situation, but still has a significant limitation: it reacts only after the problem has occurred. This means that the damage has often already begun by the time the system intervenes.

For this reason, we wish to design a more intelligent decision-making policy π_θ , parameterised by θ , which takes multiple objectives into account simultaneously. In particular, we wish to maximise the following function:

$$J(\pi_\theta) = \mathbb{E} \left[\underbrace{\sum_{t=1}^T r_t}_{\text{performance}} + \lambda_e \cdot \underbrace{E_t}_{\text{eff.}} + \lambda_s \cdot \underbrace{S_t}_{\text{stability}} + \lambda_\tau \cdot \underbrace{\tau_t}_{\text{trust}} \right] \quad (1)$$

This function represents what we wish to achieve from the system:

- r_t : good performance (e.g. high agricultural yield);
- E_t : efficient use of resources, such as water and energy;
- S_t : stability of the system over time;
- $\tau_t \in [0, 1]$: level of user trust in the system.

The parameters λ_e , λ_s and λ_τ serve to balance the importance of these objectives.

In addition to this, we also introduce a measure of resilience $\mathcal{R}$ (defined in Section 3.4), which allows us to assess how well the system manages to react and recover after a shock.

In summary, our aim is to design a system that not only functions well under normal conditions, but is also capable of adapting rapidly when the environment changes or unexpected events occur.

3.2. System Architecture

The proposed architecture consists of five types of agents that communicate with each other via asynchronous messages, according to the FIPA-ACL standard. Figure 1 provides an overview of the system and how agents interact with each other and with the environment.

The system is organized into three main levels. The first is the **reactive** level, which includes the Sensor and Farm agents. These agents operate in direct contact with the physical environment and must respond rapidly to changes, for example by reading data from sensors or activating irrigation and ventilation.

The second is the predictive and decision-making level, which comprises the Climate Agent and the Coordinator Agent. The former is responsible for generating forecasts and contextual information, whilst the latter plays a central role: it coordinates the other agents, makes more complex decisions and provides explanations for the actions taken.

Finally, the third level is that of supervision, which includes the Human Agent. This represents the human user, who can observe the system's behaviour, receive explanations and intervene when necessary, for example by modifying or cancelling certain decisions.

It is useful to make explicit what we treat as an *agent* and what we treat as a *non-agent* entity, following the classical distinction between agents, artifacts and environment adopted in the Agents & Artifacts meta-model and in BDI platforms [4, 11]. An entity is modelled as an agent only when it has its own goals, an internal deliberation step and autonomy in deciding whether and how to act; it is modelled as an artifact when it is a passive, function-providing resource that agents use or observe but that does not deliberate. Under this criterion the Sensor and Farm components are, strictly speaking, closer to artifacts: each wraps a physical resource (a sensor or an actuator), exposes a typed

Resilient Multi-Agent System Architecture

Environmental monitoring & climate adaptation

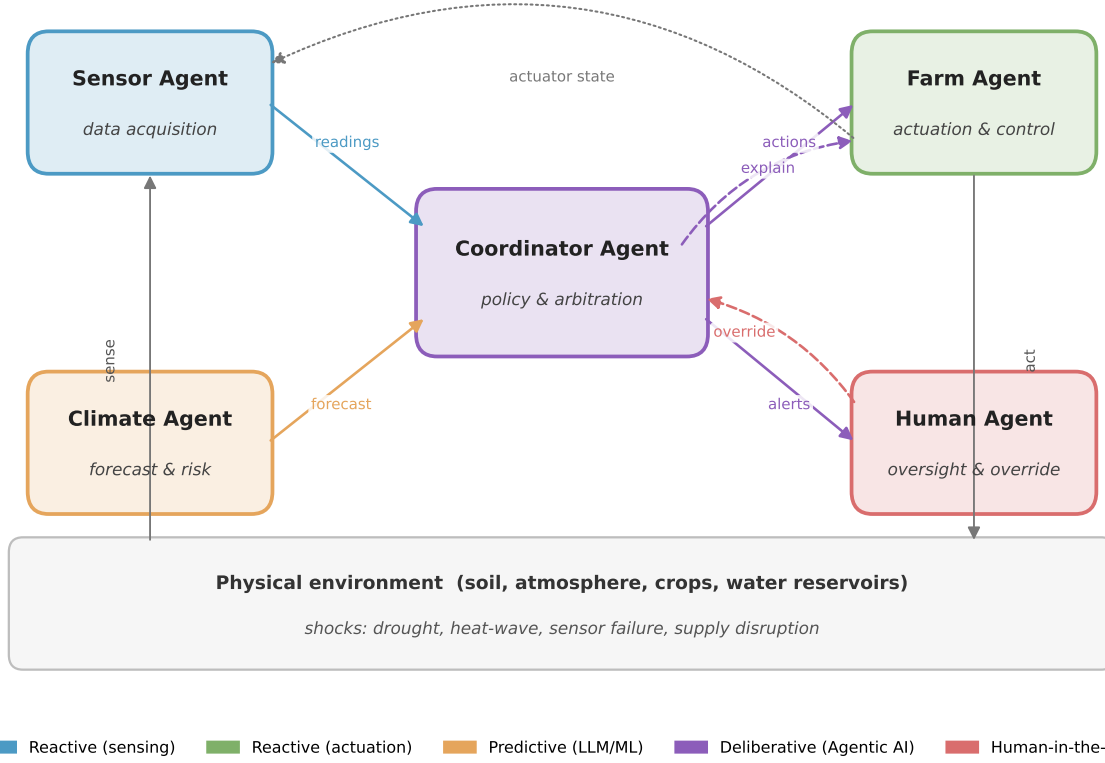


Figure 1: Resilient Multi-Agent System architecture for environmental monitoring and climate adaptation. Solid arrows denote standard message flow; dashed arrows denote out-of-band channels (override and explanation); dotted arrows denote feedback signals about actuator state. The shaded band at the bottom represents the physical environment in which shocks originate.

interface, and performs only local, non-deliberative processing such as calibration, noise filtering or safety clamping. We nonetheless realise them as thin reactive agents so that every entity shares the same $\langle \text{observe, decide, act, learn, explain} \rangle$ interface and can participate uniformly in message passing and in role re-assignment after a failure. The climate and the farm plot themselves are not agents but part of the environment in which the deliberative agents (Climate and Coordinator) and the supervisory Human Agent operate: the Climate Agent is a deliberative agent that predicts the environment, and is distinct from the climate process it forecasts. This thin-agent realisation of the reactive layer is an engineering choice that favours uniformity and resilience, and is fully compatible with an artifact-based reading of sensors and actuators.

The agents communicate with one another by exchanging structured messages (for example, to inform, request or propose actions). Each message contains not only the information, but also a confidence value $\kappa \in [0, 1]$, which indicates how certain the agent is about what it is communicating, and an explanation element ξ , which is used to make the decisions taken understandable.

An important aspect of the architecture is its modularity. All agents follow the same basic structure, which includes observation, decision-making, action, learning and explanation functions. This makes the system flexible: it is possible to add new agents, replace them or temporarily deactivate them without compromising overall operation. This feature is fundamental to ensuring the system's resilience, for

Table 1

Agent classes and their roles.

Agent	Style	Input	Output	Failure mode
Sensor	Reactive	physical signal	readings, q	degraded, failed
Farm	Reactive	action request	realised action	refuse if unsafe
Climate	Predictive	exogenous + history	forecast, warning	low-confidence
Coordinator	Deliberative	all of the above	action, explanation	escalate to human
Human	Supervisory	dashboard	override / accept	latency

example in the event of sensor failures or sudden changes in the environment.

3.3. Agent Design

The system consists of five types of agents, each with a specific role. Together, they work to monitor the environment, make decisions and adapt to changes.

The **Sensor Agent** represents a physical or virtual sensor. Its behaviour is simple and responsive: it reads data and assesses its quality. To do this, it uses basic calibration and a filter to reduce noise. It outputs not only the measured value but also a quality indicator q , which can be ok, degraded or failed. If the sensor stops working correctly, it reports the problem to the system, which can respond by reorganising the roles of the other agents.

The **Farm Agent** is connected to an actuator, such as a valve, a pump or a ventilation system. It receives commands and executes them, but with one important constraint: it never performs dangerous actions. For example, it will not open a valve beyond the permitted limits, even if requested. After performing an action, it informs the system of what it has done and how much energy it has consumed. This agent is also reactive, but designed to be safe and reliable.

The **Climate Agent** has a different role: it does not merely react, but seeks to predict what will happen. It uses historical data and weather forecasts to estimate the future evolution of the climate context, producing a probabilistic forecast:

$$\hat{P}(\mathbf{c}_{t+1:t+H} \mid \mathbf{c}_{1:t})$$

When it forecasts significant events, such as a heatwave, it sends a warning to the system. This agent allows problems to be anticipated rather than merely reacting after they have occurred.

The **Coordinator Agent** is the decision-making heart of the system. This is where the ‘smartest’ part takes place: it gathers all the information from the other agents, assesses the situation and decides what to do. It uses both a learned policy

$$\pi_{\theta} : \mathcal{S} \rightarrow \Delta(\mathcal{A})$$

and safety rules that prevent risky actions. Furthermore, it manages the resilience cycle and generates explanations for the decisions made, which are then displayed to the user.

Finally, the **Human Agent** represents the human user. Through an interface, the user can view the system’s status, the proposed decisions and the corresponding explanations. If necessary, they can intervene: accepting, rejecting or modifying the suggested actions. This human intervention is also important for updating the level of trust in the system.

In general, agents are designed to work together in a coordinated yet flexible manner: some react quickly, others predict the future, and still others make decisions or supervise the system.

Table 1 summarises the agent roles, decision style, and input/output channels.

3.4. The Resilience Loop

To enable the system to adapt to changes, we introduce the concept of the *Resilience Loop*, i.e. a continuous cycle that allows the system to monitor the environment, identify problems, react and

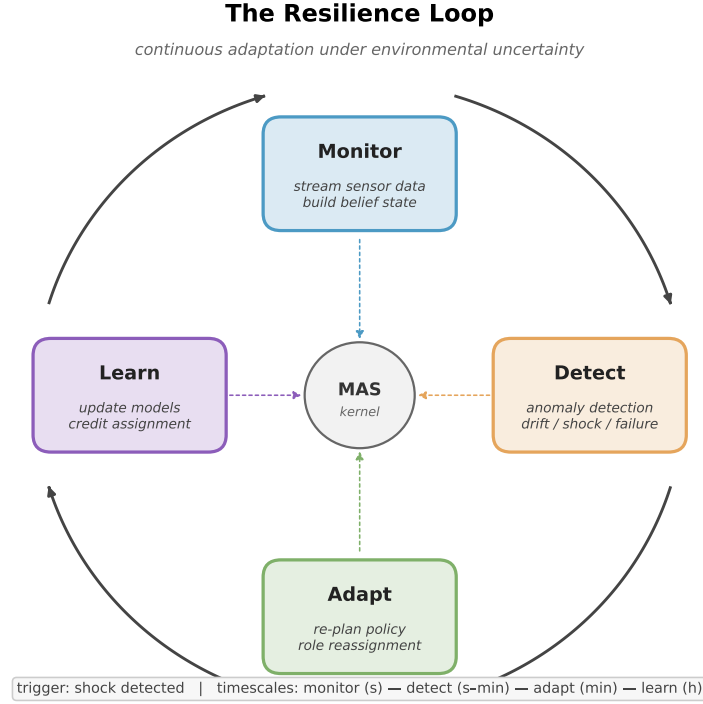


Figure 2: The Resilience Loop. The four phases (Monitor, Detect, Adapt, Learn) operate at different timescales and feed information back to the MAS kernel through the dashed inner arrows. Cyclic outer arrows represent the dominant clockwise flow under normal operation.

improve over time. This cycle is managed by the Coordinator Agent and consists of four main phases: *Monitor*, *Detect*, *Adapt* and *Learn*. Figure 2 shows its general structure.

The Resilience Loop can be positioned with respect to the classical BDI reasoning cycle (sense–deliberate–act) that underlies AgentSpeak-style platforms [10, 11] and the perceive–reason–act loop of rational agents more generally [4, 30]. It is not a replacement for that cycle but an outer control loop wrapped around it: each Coordinator deliberation step is still a standard reasoning cycle, in which the agent perceives incoming messages, updates its beliefs, selects an intention and acts. On top of ordinary perception, the *Monitor* and *Detect* phases add an explicit drift- and shock-detection stage, while *Adapt* and *Learn* add, respectively, a shock-conditioned re-planning stage and a slow policy-update stage that the classical cycle leaves implicit. Whereas the standard reasoning cycle assumes a stationary environment and reacts within a single deliberation, the Resilience Loop treats non-stationarity as a first-class concern: it operates at two timescales (a fast per-step reasoning cycle and a slow per-episode learning step) and treats recovery after a shock as an explicit, measurable objective (Eq. 2). The contribution therefore lies not in the cyclic structure itself, which is standard in MAS, but in coupling it with explicit resilience metrics and a learning-based recovery phase.

Monitor. In this phase, the system continuously observes what is happening. The Coordinator collects data sent by the other agents and updates its internal state, denoted by b_t . To account for uncertainty, it uses simple yet effective techniques, such as moving averages or probabilistic filters, which allow for the handling of noisy or incomplete data.

Detect. Once the state has been updated, the system checks whether anything is amiss. In particular, it compares the current situation with the ‘normal’ (reference) one. If the difference exceeds a certain threshold δ , a shock is detected. This may be a local problem (such as a faulty sensor) or a wider event (such as a drought or a heatwave).

Adapt. When a problem is detected, the system reacts. The Coordinator recalculates the actions to be taken using a policy adapted to the type of shock. For example, if a sensor stops working, the system can use data from nearby sensors or estimate the missing values. If, on the other hand, an extreme weather event is forecast, it can take preventive measures, such as reducing water usage to conserve it.

Learn. Once the situation returns to normal, the system analyses what has happened. At this stage, it updates its parameters to improve future decisions and saves information about the event in memory, so as to recognise similar situations more quickly in the future.

3.4.1. Resilience metric

To measure how well the system reacts to shocks, we use a resilience metric $\mathcal{R}(t)$, defined as:

$$\mathcal{R}(t) = \frac{\Phi(t) - \Phi(t_d)}{\Phi(t_0) - \Phi(t_d)}, \quad t_d \leq t \leq t_f \quad (2)$$

where $\Phi(t)$ represents the system's performance over time (in our case, yield divided by water used). The value t_0 indicates the state prior to the shock, t_d the moment of greatest difficulty, and t_f the moment at which the system has fully recovered.

The metric is normalized so that it equals 0 at the moment of greatest difficulty t_d and 1 when performance returns to its pre-shock level $\Phi(t_0)$; it is therefore typically in $[0, 1]$ during recovery but is *not* bounded above, and values greater than 1 indicate that the system ends up performing better than before the shock:

- $\mathcal{R} = 0$ means that the system fails to recover,
- $\mathcal{R} = 1$ indicates a complete recovery,
- values greater than 1 indicate that the system improves compared to the initial situation.

3.4.2. Algorithm

Algorithm 1 summarises how the Resilience Loop operates within the Coordinator Agent. In practice, the system always follows the same pattern: it observes, checks for problems, reacts if necessary, and finally learns from the experience. This process integrates with the system's normal decision-making cycle (observe, decide, act), allowing for continuous adaptation over time.

3.5. Trust and Explainability

The system's decisions are not abstract: they affect real resources, such as water and energy, and therefore also the people who use them. For this reason, it is essential that the user understands what the system is doing. If the decisions are unclear, the user will tend to ignore them or disable the system.

To avoid this problem, we have integrated a mechanism for trust and explainability directly into the agents' operation.

When the Coordinator proposes an action $\mathbf{a}_t$, it does not merely decide what to do, but also attempts to explain why. To do this, it analyses which pieces of information most influenced the decision (e.g. temperature, humidity or weather forecasts) and assigns each a 'weight' indicating how important it was. This provides a local explanation of the decision.

However, numbers and weights alone are not always easy to understand. For this reason, the system transforms this information into a brief explanation in natural language. For example, instead of showing only numerical values, it might say something like: "watering has been reduced because a heatwave is forecast and the humidity level is already low". This makes the decisions much more understandable for the user.

The user, via the Human Agent, can always intervene. They can accept the proposed decision, reject it (in which case the system performs a default safe action), or modify it within certain limits. Every interaction is recorded along with the explanation and a confidence level expressed by the user.

Algorithm 1 The Resilience Loop (Coordinator Agent).

```
1: Initialise belief  $b_0$ , baseline  $\mathcal{B}$ , policy  $\pi_\theta$ , episodic memory  $\mathcal{M}$ 
2: for  $t = 1, 2, \dots, T$  do
3:    $\mathbf{o}_t \leftarrow \text{COLLECT-INFORM-MESSAGES}()$  ▷ Monitor
4:    $b_t \leftarrow \text{BELIEF-UPDATE}(b_{t-1}, \mathbf{o}_t)$ 
5:    $d_t \leftarrow \text{WASSERSTEIN}(b_t, \mathcal{B})$  ▷ Detect
6:   if  $d_t > \delta$  then
7:      $\sigma_t \leftarrow \text{CLASSIFY-SHOCK}(b_t, \mathcal{M})$ 
8:      $\pi_\theta^\sigma \leftarrow \text{CONDITION}(\pi_\theta, \sigma_t)$  ▷ Adapt
9:     if  $\sigma_t = \text{SENSOR-FAILURE}$  then
10:       $\text{REASSIGN-ROLES}(b_t)$ 
11:     end if
12:      $\mathbf{a}_t \leftarrow \pi_\theta^\sigma(b_t)$ 
13:   else
14:      $\mathbf{a}_t \leftarrow \pi_\theta(b_t)$ 
15:   end if
16:    $\xi_t \leftarrow \text{EXPLAIN}(b_t, \mathbf{a}_t)$  ▷ Trust layer
17:    $\text{SEND-REQUEST}(\mathbf{a}_t, \xi_t)$  to Farm and Human
18:    $r_t \leftarrow \text{OBSERVE-REWARD}()$ 
19:   if end-of-episode then
20:      $\theta \leftarrow \text{UPDATE}(\theta, r_{1:t}, b_{1:t}, \mathbf{a}_{1:t})$  ▷ Learn
21:      $\mathcal{M} \leftarrow \mathcal{M} \cup \{(\sigma_t, b_{1:t}, \mathbf{a}_{1:t})\}$ 
22:   end if
23: end for
```

This information is then used by the system to improve over time. In practice, the system learns to favour decisions that have been accepted by the user in the past, thereby increasing the overall confidence level.

Finally, this approach follows the principles of trustworthy AI: the system is under human control, is transparent in its decisions, strives to be robust and safe, and keeps a record of all actions and their explanations. The aim is to build a system that is not only effective, but also reliable and understandable.

3.6. Simulation Setup

3.6.1. Scenario

We instantiate the architecture in a smart-agriculture scenario: a single greenhouse plus open-field plot growing tomatoes, equipped with six soil-moisture sensors, two temperature/humidity sensors, one CO₂ sensor, four drip-irrigation valves, and one ventilation system. The simulated season runs for 60 days with a 1-hour control cycle, giving $T = 1440$ steps per episode. We use agro-climatic data inspired by Mediterranean conditions and inject shocks at predefined intervals.

3.6.2. Shock injection

Four shock classes are injected, each in a different episode and in mixed combinations:

- **Drought:** rainfall set to zero and reference evapotranspiration raised by 50% for 15 consecutive days.
- **Heat-wave:** ambient temperature raised by 4–6 °C for 5 days.
- **Sensor failure:** two soil-moisture sensors silently output a constant reading for 48 hours.
- **Supply disruption:** water input is capped at 40% of nominal for 7 days.

3.6.3. Baselines

We compare R-MAS against two baselines.

- **Static rule-based MAS:** each Farm Agent uses a fixed threshold on its corresponding sensor; no Climate Agent, no Coordinator-level planning, no Human Agent feedback.
- **Reactive MAS (no learning):** Sensor and Farm agents plus a Coordinator that uses a hand-coded policy (no MARL, no Resilience Loop, no explanations).

These two baselines are chosen as controlled lesions of the full system rather than as competing state-of-the-art controllers: the static baseline removes prediction, learning and oversight at once, while the reactive baseline restores coordination but still omits the Resilience Loop and the learned policy. Together with the per-component ablation reported in Section 4 (R-MAS minus the Climate Agent, minus the learning phase, and minus the explanation module), this lets us attribute each gain to a specific mechanism rather than to the system as a whole. A comparison against a stronger learned baseline that shares R-MAS’s MARL policy but lacks the Resilience Loop is left to future work.

3.6.4. Metrics

We track four primary metrics.

- **Resource efficiency** E_t : kg of crop yield per cubic meter of water (reported daily).
- **Adaptation speed:** time to recover 90% of the pre-shock performance level $\Phi(t_0)$ in Eq. (2).
- **System stability** S_t : variance of the realised reward over a sliding 24-hour window.
- **User trust** τ_t : average Likert-scale score (rescaled to $[0, 1]$) reported by simulated operators after each interaction. Operators are modelled with three behavioural profiles (cautious, balanced, expert) calibrated against published survey data; we report results averaged across the three.

We also report the resilience metric $\mathcal{R}$ from Eq. (2) on the drought episode.

3.6.5. Implementation

Agents are implemented in Python on top of the Mesa agent-based simulation library. The deliberative behaviour of the Coordinator is specified in *AgentSpeak* [11], the agent-oriented programming language executed by the Jason engine of the JaCaMo platform; the AgentSpeak plans are executable and are reported in the paper in a compact, readable notation, while the Resilience Loop of Algorithm 1 is the abstract specification they realise. The numeric routines invoked by the plans (belief update, shock classification, the learned policy and the explanation step) are provided through Python action bindings.

The learning component of the *Adapt* step uses cooperative Multi-Agent Reinforcement Learning under the centralised-training, decentralised-execution paradigm. We adopt **QMIX** [16], which suits a cooperative team reward and a discrete action space (irrigation and ventilation levels) and provides a monotonic value factorisation that preserves decentralised per-agent execution. Each agent’s utility network is a small recurrent multi-layer perceptron: the local observation and previous action feed two hidden layers of 64 units with ReLU activations, followed by a GRU cell of 64 units that summarises the interaction history, and a linear output layer with one value per discrete action. The mixing network is a two-layer hypernetwork with 32 hidden units whose generated weights are constrained to be non-negative to maintain monotonicity. Training uses the Adam optimiser (learning rate 5×10^{-4}), discount factor $\gamma = 0.99$, a replay buffer of 5000 episodes, batch size 64, and a target network updated every 200 steps, with ϵ -greedy exploration annealed from 1.0 to 0.05.

The three forms of uncertainty introduced in Section 3 are handled by distinct mechanisms. *Aleatoric* uncertainty, arising from weather and sensor-noise variability, is represented by the predictive distribution of the Climate Agent, implemented as a deep ensemble of $M = 5$ forecasting networks whose spread provides a calibrated variance. *Epistemic* uncertainty, due to imperfectly known soil and crop parameters and to partial observability, is tracked in the *Monitor* phase by a particle filter with

Listing 1: Executable AgentSpeak plan run by the Coordinator (Jason). It fires when a drought shock is detected and the forecasted heat index is high, conditioning the policy and emitting an explanation.

```
// belief base updated by Monitor/Detect
+shock(drought)[source(detector)] : forecast(heat_index, H) & H > 0.7
  <- .print("drought + heat: conserving water");
    conditioned_action(drought, A);      // Adapt: QMIX policy
    explain(A, X);                        // Trust layer
    .send(farm, achieve, do(A));
    .send(human, tell, proposal(A, X)).

+override(A2)[source(human)]             // human-in-the-loop
  <- .send(farm, achieve, do(A2));
  +trust_feedback(accepted).
```

Listing 2: Python harness comparing the three systems under identical shocks and seeds.

```
systems = {"static": StaticRuleMAS(),
          "reactive": ReactiveMAS(),
          "rmas": ResilientMAS(policy=qmix)}

results = {name: [] for name in systems}
for seed in range(30):
    for name, sys in systems.items():
        env = Greenhouse(seed=seed,
                          shocks=SHOCK_SCHEDULE)
        log = run_episode(sys, env, T=1440)
        results[name].append(
            {"eff": efficiency(log),
             "recovery": recovery_time(log),
             "stability": stability(log),
             "R": resilience_R(log)})
report_mean_std(results) # Figs. 3-5, Table 2
```

$N = 1024$ particles that maintains the belief b_t ; its weighted variance is the quantity compared against the baseline in the *Detect* phase, using a Wasserstein distance with threshold $\delta = 0.18$. *Structural shocks* are handled outside the probabilistic model, through the shock-classification and role-reassignment logic of the Resilience Loop. The complete network topologies and hyperparameters are listed in the released configuration files. The MARL training uses PyTorch and is performed offline on 200 episodes of $T = 1440$ steps each. We run 30 independent test seeds per configuration and report mean and standard deviation. All experiments are reproducible with the configuration files released with the paper.

To make the experimental setup concrete, Listing 1 reports a representative executable AgentSpeak plan of the Coordinator, run by the Jason engine, and Listing 2 shows the Python loop used to compare R-MAS against the two baselines on the same shock schedule and seeds.

4. Results and Discussion

The results in Figure 3 clearly show that the proposed system (R-MAS) performs better than traditional methods in all the situations considered. During a 60-day simulation with a 15-day drought, the fixed-rule-based system uses too much water at the start of the shock and sees its efficiency drop from around 0.62 to less than 0.40.

A reactive multi-agent system manages to improve the situation slightly, but reacts too late, when the problem is already underway. In contrast, R-MAS manages to anticipate the shock thanks to the Climate Agent’s forecasts, maintaining efficiency at around 0.65 and recovering rapidly after the crisis, even achieving slightly better performance than before (resilience $\mathcal{R} \approx 1.04$). The curves in Figure 3 report the mean over the 30 test seeds; the shaded bands, where shown, give one standard deviation, which stays narrow (< 0.03) outside the shock window and widens during recovery, confirming that

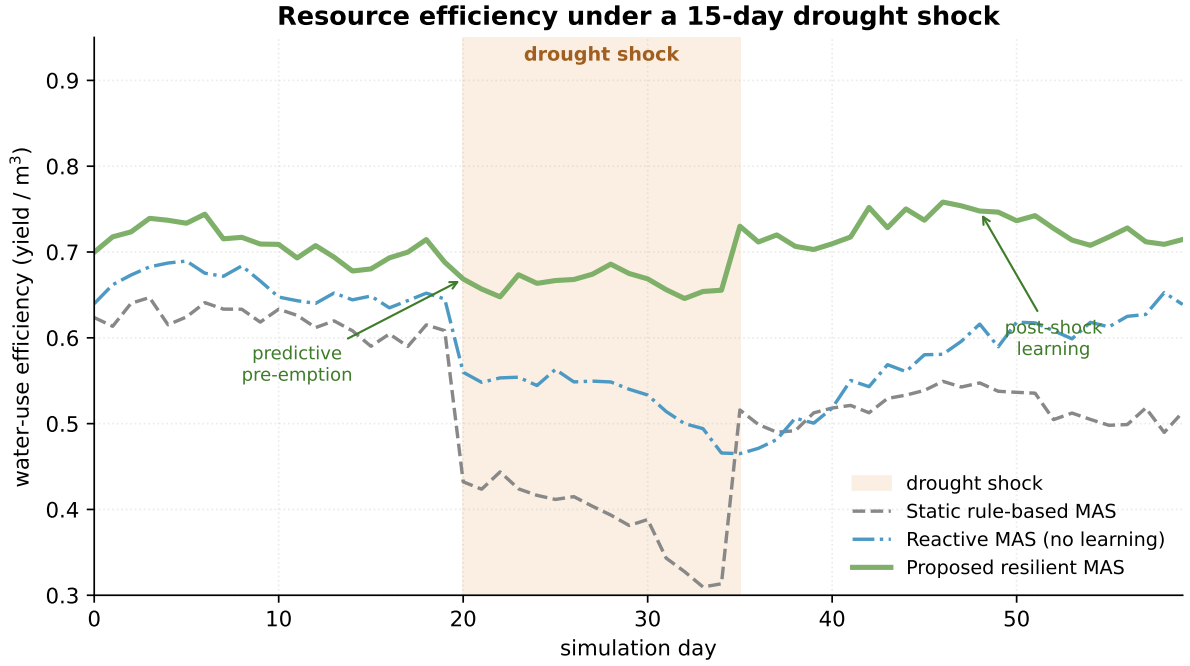


Figure 3: Daily water-use efficiency over a 60-day simulated season containing a 15-day drought shock (shaded region). The proposed R-MAS (green) anticipates the shock thanks to the Climate Agent and recovers to a higher post-shock efficiency, while the static rule-based baseline (grey) collapses and the reactive baseline (blue) recovers only partially.

the ordering of the three systems is not an artefact of a single seed. This behaviour is also observed for other types of events in Figure 4: R-MAS reduces recovery times by a factor of 2 to 3.5 compared to the static system and by about a factor of 2 compared to the reactive system, with the best result in cases of sensor failure (approximately 5.2 hours versus 18.4). System stability also improves significantly as shown in Table 2: whilst the static system exhibits strong fluctuations, particularly when data is unreliable, R-MAS maintains stable values across all scenarios (for example, 0.84 versus 0.41 in the case of drought).

Table 2

System stability (S_t , higher is better) across shock classes. Values are means across 30 seeds; standard deviations are in parentheses.

Shock	Static	Reactive	R-MAS (ours)
Drought	0.41 (.05)	0.62 (.04)	0.84 (.03)
Heat-wave	0.48 (.06)	0.66 (.04)	0.86 (.03)
Sensor failure	0.32 (.07)	0.55 (.05)	0.81 (.04)
Supply disruption	0.45 (.05)	0.63 (.04)	0.83 (.03)

Another important aspect is user confidence (Figure 5): without explanations, confidence tends to decline and does not recover after a shock; adding explanations improves it, but it is above all the possibility of human intervention that makes the difference, raising confidence above 0.90 and resulting in a 35–40% improvement. The experiments also show that every component of the system is important: without the Climate Agent, it is not possible to anticipate events; without the learning phase, performance deteriorates over time; and without the explanation module, trust declines even if performance remains good. Table 3 quantifies this component ablation on the drought episode: removing the Climate Agent erases the predictive pre-emption and lowers the resilience score the most, removing the learning phase mainly hurts post-shock efficiency, and removing the explanation module

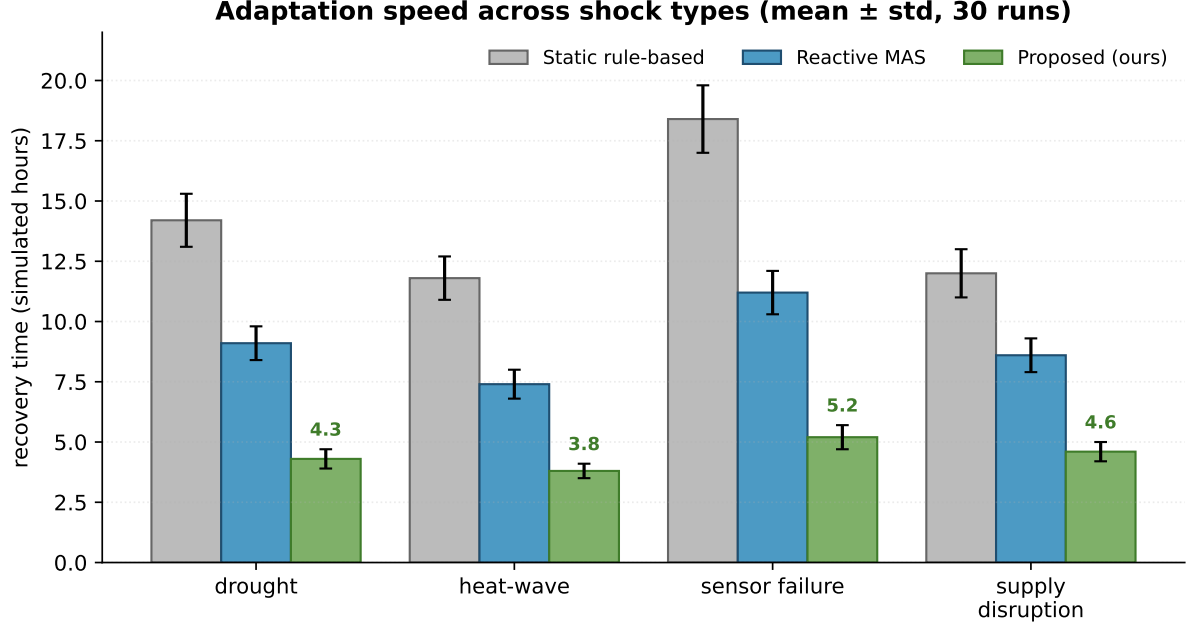


Figure 4: Recovery time after each of the four shock classes (mean $\pm$ standard deviation across 30 seeds). The proposed R-MAS consistently achieves the lowest recovery time, with the largest margin on sensor failures.

leaves efficiency almost unchanged while collapsing trust, confirming that performance and trust are driven by different parts of the architecture. In general, these results suggest that resilience stems from a combination of rapid response, predictive capability and coordinated decisions, rather than from a single component. However, the work has some limitations: the results are obtained from simulation, the confidence model is based on simulated users, and the quality of decisions also depends on the accuracy of weather forecasts. Since no real users were involved, the reported trust gains should be interpreted as evidence about the underlying mechanism, namely the combination of explanation and human override, rather than as a validated estimate of the trust of real operators; the human-in-the-loop results obtained with simulated operator profiles are therefore preliminary, and a study with real users on real-world data is needed before drawing conclusions about field-level trustworthiness. Furthermore, the learned policy may perform sub-optimally if the objectives are poorly defined. Despite this, the proposed architecture is general and can also be applied to other contexts, such as energy networks, water systems or urban monitoring, whilst maintaining the same agent-based approach, resilience and explainability.

Table 3

Component ablation on the drought episode (means across 30 seeds). Each row removes one module from the full R-MAS. E is water-use efficiency during the shock, $\mathcal{R}$ the resilience score (Eq. 2), and τ the final user-trust score; higher is better in all columns.

Configuration	E	$\mathcal{R}$	τ
Full R-MAS (ours)	0.65	1.04	0.92
– no Climate Agent	0.54	0.79	0.88
– no learning (<i>Learn</i>)	0.58	0.86	0.90
– no explanation module	0.64	1.02	0.61

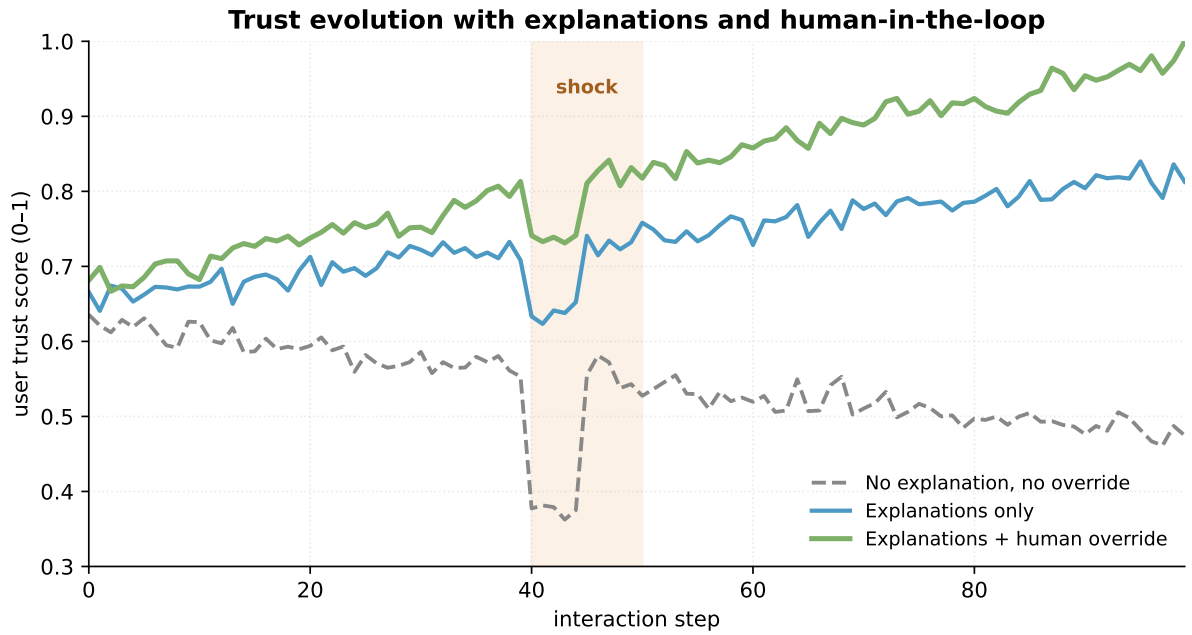


Figure 5: Trust evolution across 100 interaction steps with a shock at step 40. Explanations alone improve the post-shock recovery; the combination of explanations and human override yields the highest and most stable trust.

5. Conclusion and Ethical Considerations

In this paper, we have presented a resilient multi-agent system for environmental monitoring and adaptation to climate change, which combines reactive agents, AI-based predictive components and a human control layer. The results show that the system reacts better to environmental shocks than traditional solutions: it reduces recovery times by a factor of about 2–3.5, maintains good efficiency even during critical events such as prolonged drought and, with simulated operators, substantially increases user confidence (by up to 35–40%) once clear explanations and human intervention are available. In addition to performance, we have also focused on ethical considerations. In particular, it is important to ensure that the user retains real control over the system, avoiding excessive reliance on automated decisions; for this reason, every critical action can be checked or modified. We have also considered the issue of inequality in access to technology, given that advanced systems could favour larger companies over small-scale farmers. Another aspect concerns the environmental impact of AI itself: training and using the models require computational resources, which must be balanced against the benefits gained. From a privacy perspective, sensitive data remains local and is not shared directly. Finally, some challenges remain, such as system security in the event of attacks and the fact that the results were obtained in simulation. In the future, it will be important to test the system in real-world contexts, improve integration with more accurate models and directly involve users in the design process, to make the system even more useful, reliable and sustainable.

Acknowledgments

This work was partially supported by PNRR MIUR Project PE0000013-FAIR. The FAIR project is committed to promoting an advanced vision of Artificial Intelligence, driving research and development in this crucial field and constantly keeping ethical, legal and sustainability considerations in mind.

Declaration on Generative AI

During the preparation of this work, the authors used generative AI tools only to a very limited extent, for minor language editing. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] IPCC, Climate Change 2022: Impacts, Adaptation and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change, Technical Report, Intergovernmental Panel on Climate Change, Cambridge, UK, 2022.
- [2] S. Wolfert, L. Ge, C. Verdouw, M.-J. Bogaardt, Big data in smart farming – a review, *Agricultural Systems* 153 (2017) 69–80. doi:10.1016/j.agry.2017.01.023.
- [3] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, D. Bochtis, Machine learning in agriculture: A review, *Sensors* 18 (2018) 2674. doi:10.3390/s18082674.
- [4] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed., John Wiley & Sons, Chichester, UK, 2009.
- [5] P. Stone, M. Veloso, Multiagent systems: A survey from a machine learning perspective, *Autonomous Robots* 8 (2000) 345–383. doi:10.1023/A:1008942012299.
- [6] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, ReAct: Synergizing reasoning and acting in language models, in: *International Conference on Learning Representations (ICLR)*, 2023.
- [7] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, M. S. Bernstein, Generative agents: Interactive simulacra of human behavior, in: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2023. doi:10.1145/3586183.3606763.
- [8] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, et al., A survey on large language model based autonomous agents, *Frontiers of Computer Science* 18 (2024) 186345. doi:10.1007/s11704-024-40231-1.
- [9] M. Wooldridge, N. R. Jennings, Intelligent agents: theory and practice, *The Knowledge Engineering Review* 10 (1995) 115–152. doi:10.1017/S0269888900008122.
- [10] A. S. Rao, M. P. Georgeff, BDI agents: From theory to practice, in: *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco, CA, USA, 1995, pp. 312–319.
- [11] R. H. Bordini, J. F. Hübner, M. Wooldridge, *Programming multi-agent systems in AgentSpeak using Jason*, in: *Wiley Series in Agent Technology*, Wiley, 2007.
- [12] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, et al., Language models are few-shot learners, in: *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020, pp. 1877–1901.
- [13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017, pp. 5998–6008.
- [14] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, et al., Retrieval-augmented generation for knowledge-intensive NLP tasks, in: *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020, pp. 9459–9474.
- [15] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, I. Mordatch, Multi-agent actor-critic for mixed cooperative-competitive environments, in: *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017, pp. 6379–6390.
- [16] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, S. Whiteson, QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning, in: *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 4292–4301.
- [17] C. S. Holling, Resilience and stability of ecological systems, *Annual Review of Ecology and Systematics* 4 (1973) 1–23. doi:10.1146/annurev.es.04.110173.000245.

- [18] B. Walker, C. S. Holling, S. R. Carpenter, A. Kinzig, Resilience, adaptability and transformability in social-ecological systems, *Ecology and Society* 9 (2004) 5.
- [19] C. Folke, S. R. Carpenter, B. Walker, M. Scheffer, T. Chapin, J. Rockström, Resilience thinking: integrating resilience, adaptability and transformability, *Ecology and Society* 15 (2010) 20.
- [20] D. Henry, J. E. Ramirez-Marquez, On the impacts of power outages during Hurricane Sandy – a resilience-based analysis, *Systems Engineering* 19 (2016) 59–75. doi:10.1002/sys.21338.
- [21] High-Level Expert Group on AI, Ethics Guidelines for Trustworthy AI, Technical Report, European Commission, Brussels, 2019.
- [22] M. T. Ribeiro, S. Singh, C. Guestrin, “Why should I trust you?”: Explaining the predictions of any classifier, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144. doi:10.1145/2939672.2939778.
- [23] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017, pp. 4765–4774.
- [24] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, et al., Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, *Information Fusion* 58 (2020) 82–115. doi:10.1016/j.inffus.2019.12.012.
- [25] S. A. Huda, M. Canzaniello, D. Annunziata, F. Piccialli, Graph-based federated learning for smart healthcare: A comprehensive survey, *Future Generation Computer Systems* 183 (2026) 108524. URL: <https://www.sciencedirect.com/science/article/pii/S0167739X26001585>. doi:<https://doi.org/10.1016/j.future.2026.108524>.
- [26] F. Amato, L. Qiu, M. Tanveer, S. Cuomo, F. Giampaolo, F. Piccialli, Towards one-shot federated learning: Advances, challenges, and future directions, *arXiv preprint arXiv:2505.02426* (2025). URL: <https://arxiv.org/abs/2505.02426>. doi:10.48550/arXiv.2505.02426. arXiv:2505.02426.
- [27] D. Annunziata, P. Qi, K. Li, F. Piccialli, Towards trustworthy iiot: A reinforcement learning framework with federated unlearning, *IEEE Internet of Things Journal* (2026). doi:10.1109/JIOT.2026.3697565.
- [28] F. Tao, H. Zhang, A. Liu, A. Y. C. Nee, Digital twin in industry: State-of-the-art, *IEEE Transactions on Industrial Informatics* 15 (2019) 2405–2415. doi:10.1109/TII.2018.2873186.
- [29] C. Verdouw, B. Tekinerdogan, A. Beulens, S. Wolfert, Digital twins in smart farming, *Agricultural Systems* 189 (2021) 103046. doi:10.1016/j.agry.2020.103046.
- [30] S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2020.