applied sciences

MDPI

Article

From Statistical Filtering to Adaptive Reinforcement Learning: A Progressive Framework for IoT Time-Series Anomaly Detection

Luis Miguel Pires [1,2,3,*] **and Vitor Fialho** [2,4,*]

[1] Technologies and Engineering School (EET), Instituto Politécnico da Lusofonia (IPLuso), 1700-098 Lisbon, Portugal
[2] Department of Electronical Engineering, Telecommunications and Computers (DEETC), Instituto Superior de Engenharia de Lisboa (ISEL), 1959-007 Lisbon, Portugal
[3] School of Communication, Arts and Information (ECATI), Lusofona University, 1749-024 Lisbon, Portugal
[4] UNINOVA-CTS, NOVA University of Lisbon, Campus de Caparica, 2829-516 Monte de Caparica, Portugal
* Correspondence: luis.pires@ipluso.pt (L.M.P.); vitor.fialho@isel.pt (V.F.)

Abstract

This paper proposes a lightweight and adaptive anomaly detection framework for Internet of Things (IoT) time-series data that progressively combines statistical filtering with reinforcement learning (RL)-based decision mechanisms. Three classical statistical filters, Hampel, interquartile range (IQR), and Z-score, are initially evaluated under controlled IoT anomaly scenarios. While fixed-parameter configurations perform well under specific conditions, their performance degrades in non-stationary and heterogeneous environments. To address this limitation, a tabular Q-learning agent is introduced to dynamically select both filtering methods and their associated parameters according to scenario-specific signal characteristics. By extending the action space to include joint filter and parameter selection, the framework improves adaptability while reducing the need for manual tuning. A multi-agent reinforcement learning (MARL) formulation is further introduced to support distributed learning across heterogeneous IoT environments. The framework is additionally evaluated using real-world IoT temperature data augmented with controlled anomaly injection, enabling reproducible benchmarking under partially realistic sensing conditions. Experimental results show that both RL and MARL maintain stable detection performance across heterogeneous sensor streams. While MARL does not systematically outperform the single-agent approach in detection accuracy, it improves scalability and supports scenario-specific policy specialization, which is particularly relevant for distributed IoT deployments. Overall, the proposed approach provides a lightweight, interpretable, and computationally efficient solution for adaptive anomaly detection in resource-constrained IoT systems.

Keywords: IoT anomaly detection; time-series anomaly detection; statistical filtering; Hampel filter; interquartile range (IQR); Z-score; reinforcement learning; Q-learning; multi-agent systems; adaptive filtering; low-power IoT

Academic Editor: Vito Conforti

Received: 7 May 2026
Revised: 26 May 2026
Accepted: 1 June 2026
Published: 3 June 2026

1. Introduction

IoT systems have rapidly expanded, leading organizations to deploy cost-effective sensors for continuous monitoring of environmental conditions, smart city operations, industrial processes, and critical infrastructure. These systems generate large volumes of time-series data, which are affected by several factors, including noise, sensor performance

degradation, communication failures, and environmental disturbances. Robust anomaly detection is essential to ensure data reliability in IoT systems operating under dynamic and noisy conditions [1,2].

Anomaly detection in IoT time-series data is typically based on statistical filtering methods. These methods are well-suited for resource-constrained environments due to their simplicity, low computational cost, and interpretability, requiring minimal processing power while providing straightforward and transparent results [3,4]. However, their main limitation lies in the use of fixed parameters, typically defined manually under assumptions of stationary signal behavior. Such static settings create operational limitations, as IoT environments exhibit evolving data patterns and changing conditions that directly impact detection performance [5].

To address these limitations, learning-based approaches have been increasingly explored. Supervised Machine Learning (ML) techniques such as Support Vector Machines (SVMs) and Random Forests (RFs) can achieve high detection performance when trained on adequately labeled datasets [6]. However, their application in IoT contexts faces several challenges, including the scarcity of labeled data, limited generalization to unseen conditions, and the need for continuous retraining as environments evolve [7].

Reinforcement learning (RL) offers an alternative paradigm in which agents learn decision-making policies through interaction with the environment, receiving feedback in the form of rewards without requiring labeled data [8]. These characteristics make RL particularly suitable for environments with uncertainty and evolving conditions. In this context, RL can be used to select filtering methods, adjust their parameters, and balance detection performance with false negative reduction [9].

This work proposes a progressive framework that combines statistical filtering with RL-based adaptive decision mechanisms for IoT anomaly detection. The architecture evolves from fixed statistical filtering to adaptive filter and parameter selection, and finally to decentralized MARL specialization across distributed sensing scenarios. Synthetic datasets were initially used to ensure controlled and reproducible evaluation, followed by stress-test experiments and real-world validation under heterogeneous IoT conditions. The proposed approach aims to bridge lightweight interpretable anomaly detection with adaptive learning capabilities while preserving low computational complexity.

The main contributions of this work can be summarized as follows:

- A systematic evaluation of classical statistical filters for IoT anomaly detection under diverse and controlled anomaly scenarios.
- The development of a reinforcement learning (RL) environment that enables adaptive filter selection using reward functions based on detection accuracy and false negative penalization.
- An extended RL framework that jointly optimizes filter selection and parameter configuration.
- The development of a multi-agent reinforcement learning (MARL) framework enabling scenario-specific policy specialization in distributed IoT environments.
- A comprehensive comparison between statistical filtering, single-agent RL, and MARL approaches.
- The design of synthetic datasets with non-stationary characteristics for multi-node evaluation, together with a MARL formulation based on implicit specialization and global reward mechanisms without direct agent-to-agent communication.

Despite recent advances, the integration of lightweight statistical filtering with adaptive reinforcement learning remains limited, particularly in multi-agent IoT scenarios. Unlike deep learning–based anomaly detection approaches that often require substantial computational resources and large training datasets, the proposed framework preserves

interpretability, low computational overhead, and adaptive behavior across heterogeneous IoT sensing conditions.

Section 2 reviews the background and related work, Section 3 presents the proposed methodology, and Section 4 reports the experimental results obtained under synthetic and real-world conditions. The main findings and limitations are discussed in Section 5, while Section 6 concludes the paper and outlines future research directions. Additional implementation details are provided in Appendices A and B.

2. Background and Related Work

Anomaly detection in IoT time-series data has been extensively studied, primarily using fixed statistical techniques and supervised learning models, which often suffer from manual parameter tuning and limited adaptability. However, classical methods such as Hampel, IQR, and Z-score filters, while widely used in resource-constrained nodes due to their low computational cost and interpretability, are highly dependent on the operating scenario. More recent research has explored reinforcement learning (RL) to enable adaptive decision-making, allowing systems to autonomously select or adjust detection strategies based on evolving data characteristics.

2.1. Background

IoT ecosystems rely on large-scale deployments of distributed sensors that continuously generate time-series data. These data streams are temporally correlated, exhibit non-stationary behavior, and are commonly affected by noise, missing samples, communication disturbances, and sensor faults. In practical deployments, anomalies may emerge due to hardware degradation, environmental interference, transmission errors, or unforeseen operational conditions, making anomaly detection a critical challenge in IoT data analytics [10].

Representative examples of the anomaly categories considered throughout this study are illustrated in Figure 1, including impulsive spikes, gradual drifts, periodic perturbations, flat corruption events, and correlated multi-node disturbances commonly observed in practical IoT sensing environments.

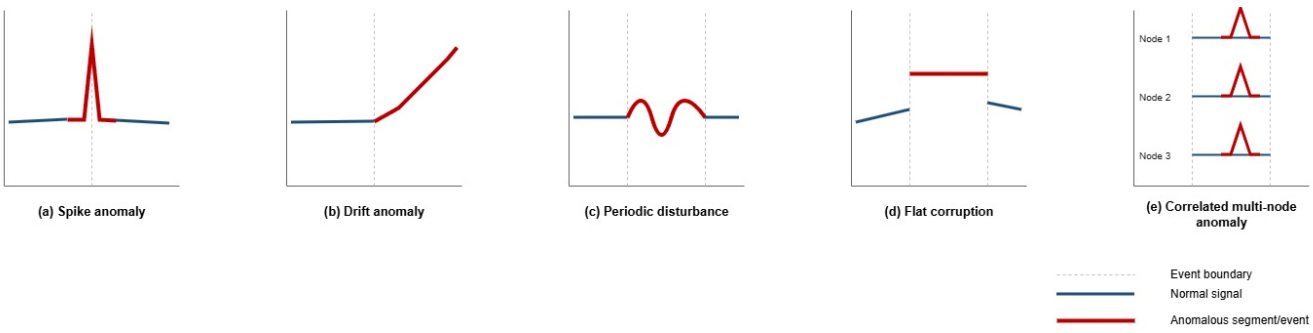

Figure 1. Representative IoT time-series anomaly types considered in this work.

Spike anomalies correspond to abrupt isolated deviations typically associated with impulsive disturbances or transient sensing faults. Drift anomalies represent gradual deviations evolving over time, often related to sensor aging or calibration degradation. Periodic perturbations may arise from cyclic interference sources or environmental effects, while flat corruption events correspond to frozen or saturated sensor readings. Correlated multi-node anomalies occur when multiple sensing nodes simultaneously exhibit synchronized abnormal behavior due to shared environmental or communication disturbances.

Classical statistical techniques such as Hampel filtering, IQR analysis, and Z-score detection remain widely used in IoT anomaly detection due to their low computational com-

plexity, interpretability, and suitability for resource-constrained embedded systems [6,11]. However, these methods typically rely on fixed thresholds and manually selected parameters, which may reduce robustness under dynamic operating conditions.

To address these limitations, RL has recently emerged as a promising adaptive decision-making framework for IoT environments [5,12]. By continuously interacting with the environment and adjusting decisions according to observed conditions and reward feedback, RL enables adaptive selection and tuning of anomaly detection strategies under evolving sensing conditions.

2.2. Related Works

Existing approaches to IoT anomaly detection can be broadly categorized into statistical and ML methods. Adhikari et al. [13] highlights the trade-off between the low computational cost and interpretability of statistical techniques and the adaptability of ML-based solutions. Statistical methods are lightweight and suitable for resource-constrained IoT devices; however, they generally require manual parameter tuning and exhibit limited robustness under dynamic operating conditions. Conversely, ML-based approaches improve adaptability and detection capability but often depend on labeled datasets, which are frequently unavailable or difficult to obtain in practical IoT deployments. Recent studies have explored RL to introduce adaptive decision-making into anomaly detection systems. Hu et al. [14] proposed a Deep Q-Network framework for adaptive policy selection, while Wali et al. [15] investigated RL-based anomaly detection in IoT communication signals under dynamically changing conditions. Although these approaches improve adaptability, they frequently rely on computationally expensive models that are difficult to deploy in low-power IoT environments. In addition, many existing RL-based solutions depend on centralized architectures or neural-network-based implementations with increased computational complexity. RL has also been investigated in distributed IoT environments. Ngo et al. [9] proposes a contextual bandit framework for adaptive detector selection in hierarchical edge systems, demonstrating that contextual model selection can improve detection accuracy while reducing computational overhead. More recently, MARL has attracted growing interest in distributed IoT applications. Diaz et al. [16] investigated MARL for intrusion detection in IoT networks, while Chen et al. [17] studied decentralized decision-making strategies for AI-enabled IoT security systems. These studies demonstrate the scalability and resilience of distributed learning approaches; however, most focus on cybersecurity and network intrusion detection rather than anomaly detection in IoT sensor time-series data.

Recent RL-based anomaly detection approaches often rely on deep neural architectures, such as DQN-based agents, autoencoder-based detectors, or hybrid RL-DL models. Deep RL approaches were intentionally avoided in order to preserve low computational complexity, explainability, and deployment feasibility in low-power IoT edge devices. These methods can improve detection flexibility, but they usually require larger training datasets, higher memory availability, and more computational resources. Such requirements may limit their direct applicability in low-power IoT sensing nodes or edge devices with restricted processing capacity. In contrast, the proposed framework does not aim to replace statistical detection with a complex black-box model. Instead, it uses tabular Q-learning as a lightweight decision layer placed above classical statistical filters. The learning agent selects the most appropriate filter and parameter configuration according to the operating condition, while the actual anomaly detection remains based on interpretable statistical mechanisms such as Hampel, IQR, and Z-score filtering. This design preserves transparency, reduces computational overhead, and allows the learned policy to be directly inspected through the Q-table. The proposed MARL extension also differs from existing

MARL-based intrusion detection or cybersecurity-oriented IoT frameworks. Rather than relying on communication-heavy cooperative agents or deep multi-agent architectures, each agent learns a local policy associated with a specific sensing condition or node context. This formulation is intentionally simple and suitable for distributed IoT deployments where communication, memory, and energy resources are limited. Therefore, the contribution of this work lies in the progressive integration of statistical filtering, tabular RL-based adaptive selection, parameter adaptation, and lightweight multi-agent specialization for IoT time-series anomaly detection. Table 1 provides a comparison between existing RL/MARL anomaly detection approaches and the proposed framework.

Table 1. Comparison between existing RL/MARL anomaly detection approaches and the proposed framework.

Approach Type	Typical Model	Main Advantage	Main Limitation	Difference from This Work
Deep RL anomaly detection	DQN/neural RL	High adaptability	Higher computational cost and lower interpretability	This work uses tabular Q-learning with explicit filter–parameter actions
RL-based IoT anomaly detection	Single-agent RL	Adaptive policy learning	Often centralized or task-specific	This work focuses on lightweight statistical filter selection
MARL intrusion detection	Distributed agents	Decentralized decisions	Mainly cybersecurity-oriented	This work targets sensor time-series anomaly detection
Statistical anomaly detection	Hampel/IQR/Z-score	Low cost and interpretable	Fixed thresholds and limited adaptability	This work adds adaptive RL-based selection and tuning
Proposed framework	Statistical filters and tabular RL/MARL	Lightweight, interpretable, adaptive	Limited to discrete actions	Designed for edge-oriented IoT deployments

Table 2 summarizes representative studies related to IoT anomaly detection and RL-based adaptive approaches, highlighting their main characteristics and limitations.

Table 2. Summary of related work on IoT anomaly detection.

Ref.	Approach	Adaptivity	Learning Paradigm	Main Limitation
[13]	Survey of statistical and ML methods	Partial	Supervised and Unsupervised	Requires labeled datasets
[14]	Deep RL anomaly detection	Yes	Single-agent RL	Complex model architecture
[15]	RL-based signal anomaly detection	Yes	Single-agent RL	Focus on communication signals
[9]	Contextual RL model selection	Yes	Single-agent RL	Limited parameter adaptation
[16]	Distributed MARL intrusion detection	Yes	MARL	Focus on network security
[17]	AI-enabled IoT security framework	Yes	MARL	Not focused on time-series anomalies

Table 2. *Cont.*

Ref.	Approach	Adaptivity	Learning Paradigm	Main Limitation
[18]	Unsupervised multivariate IoT anomaly detection	Partial	Autoencoder/DL	High computational complexity
[19]	WSN anomaly detection survey	Partial	ML-based	Limited edge deployment discussion
[20]	DRL-based IoT intrusion detection survey	Yes	Deep RL	Cybersecurity-oriented focus
[21]	Hyperlocal IoT anomaly detection	Partial	Unsupervised ML	Environmental application specificity
[22]	Multi-stage IoT anomaly detection	Partial	Hybrid ML	Increased processing overhead
This work	Statistical filters, RL and MARL	Yes	Single-agent RL and MARL	Simulation-based validation

The recent literature demonstrates growing interest in adaptive anomaly detection strategies for IoT systems, particularly through ML, deep learning (DL), and RL approaches. Survey studies highlight that classical statistical methods remain attractive for resource-constrained IoT devices due to their interpretability and low computational complexity, whereas ML and DL methods generally improve adaptability at the cost of higher computational and memory requirements [18,19]. More recent works have explored RL and Deep RL techniques for adaptive anomaly detection and intrusion detection in IoT environments. These approaches improve dynamic decision-making capability and adaptability under changing operating conditions [9,14–22]. However, many existing RL-based solutions rely on deep neural architectures, centralized learning strategies, or cybersecurity-oriented formulations that may be difficult to deploy in lightweight edge-oriented sensing environments. Although MARL has recently attracted attention in distributed IoT applications [16,17], its use for adaptive sensor time-series anomaly detection remains relatively limited. Existing approaches typically focus on intrusion detection, communication security, or computationally intensive learning frameworks. In contrast, the proposed framework combines interpretable statistical filtering with lightweight tabular RL and decentralized MARL-based specialization while maintaining suitability for low-power IoT environments [20,23,24].

3. Materials and Methods

The anomaly detection problem addressed in this study is first defined, followed by a description of the overall system architecture proposed to solve it. First, the limitations of traditional statistical methods with fixed parameters in dynamic IoT environments are discussed, motivating the need for adaptive decision mechanisms. A learning-based framework is then introduced, in which RL is employed to select detection strategies. Finally, the single-agent and multi-agent formulations adopted in the system are described.

3.1. Problem Definition

IoT-based monitoring systems generate time-series data that are frequently affected by various disturbances such as sporadic spikes, impulsive noise, drift, and sensor saturation or flat-line behavior. Such anomalies degrade data quality and may compromise downstream tasks such as forecasting, alert generation, and predictive maintenance. This work

considers IoT temperature time-series data and aims to develop an adaptive anomaly detection system capable of selecting the most appropriate filtering methods and their associated parameters to maximize detection performance while minimizing missed anomalies. The core challenge arises from the fact that no single statistical filter configuration can effectively handle all anomaly types. Different anomaly categories produce distinct statistical distortions in the signal distribution, making different filters and parameterizations preferable under different operating conditions. The solution requires an adaptive decision system which selects the optimal option according to the current operational situation. The proposed framework is organized into three progressive stages: a statistical filtering baseline, a single-agent RL adaptive selection layer, and a MARL-based distributed specialization layer. Figure 2 presents a UML class diagram of the proposed RL-based framework.

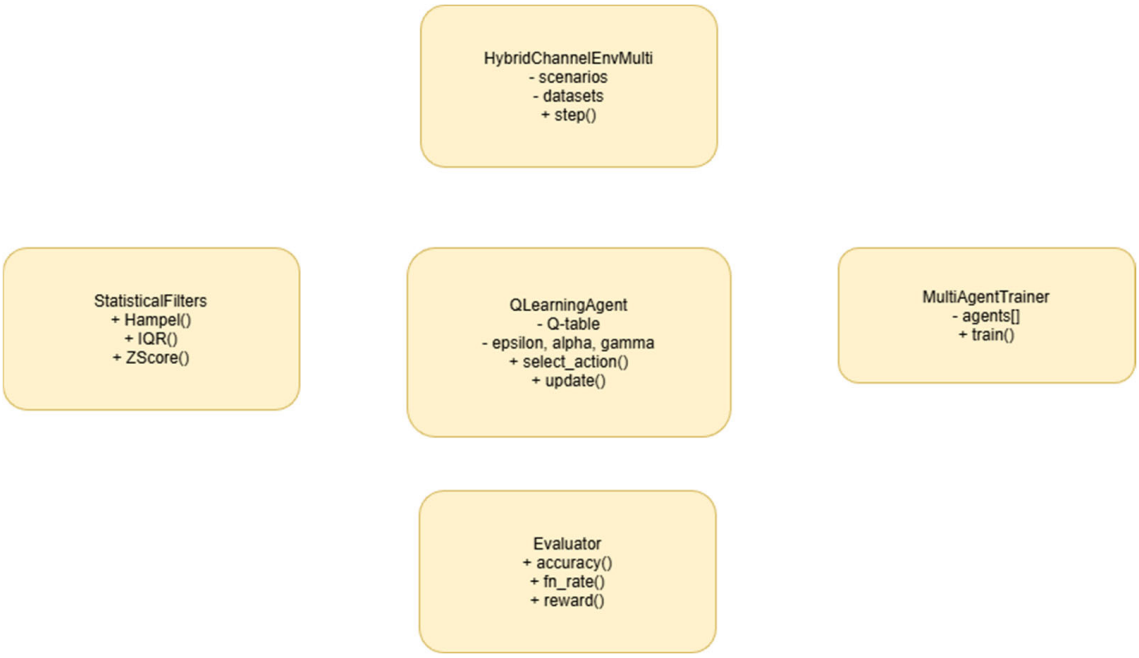

Figure 2. UML class diagram of the proposed RL-based framework.

Figure 3 illustrates the overall experimental pipeline adopted in this study. The methodology follows a progressive evaluation strategy, starting from statistical baseline detection and evolving towards adaptive RL and MARL strategies.

The experimental framework is illustrated in Figure 4, which integrates the methodological workflow with the reinforcement learning process and evaluation procedures. The left side of the diagram represents the layered architecture of the proposed approach. The process begins with Layer 0, where synthetic IoT time-series data are generated and labeled to create controlled anomaly scenarios. Layer 1 corresponds to the statistical detection baseline, where classical filters are applied using fixed parameters to establish reference performance.

In Layer 2, a reinforcement learning agent is introduced to enable adaptive filter selection based on observed conditions. Layer 3 extends this approach by allowing joint optimization of filter selection and parameter configuration. Finally, a multi-agent reinforcement learning (MARL) layer enables scenario-specific policy specialization through independent agents.

The central part of the diagram represents the RL interaction loop, where the agent observes the environment through a state representation, selects an action, receives a reward based on detection performance, and updates its policy using the Q-learning algorithm.

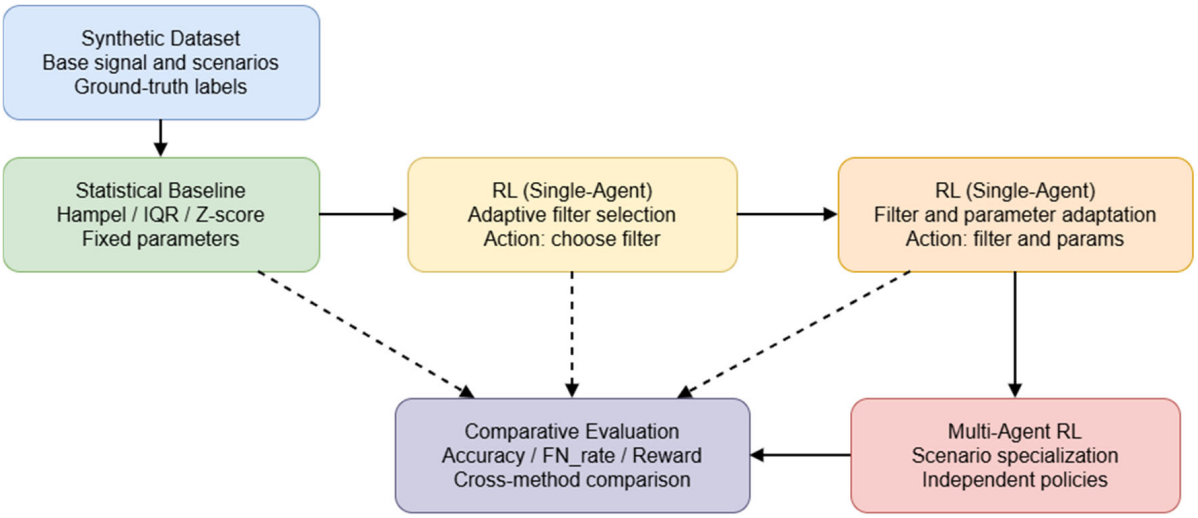

Figure 3. Experimental pipeline for progressive evaluation of statistical baselines and RL-based adaptive anomaly detection.

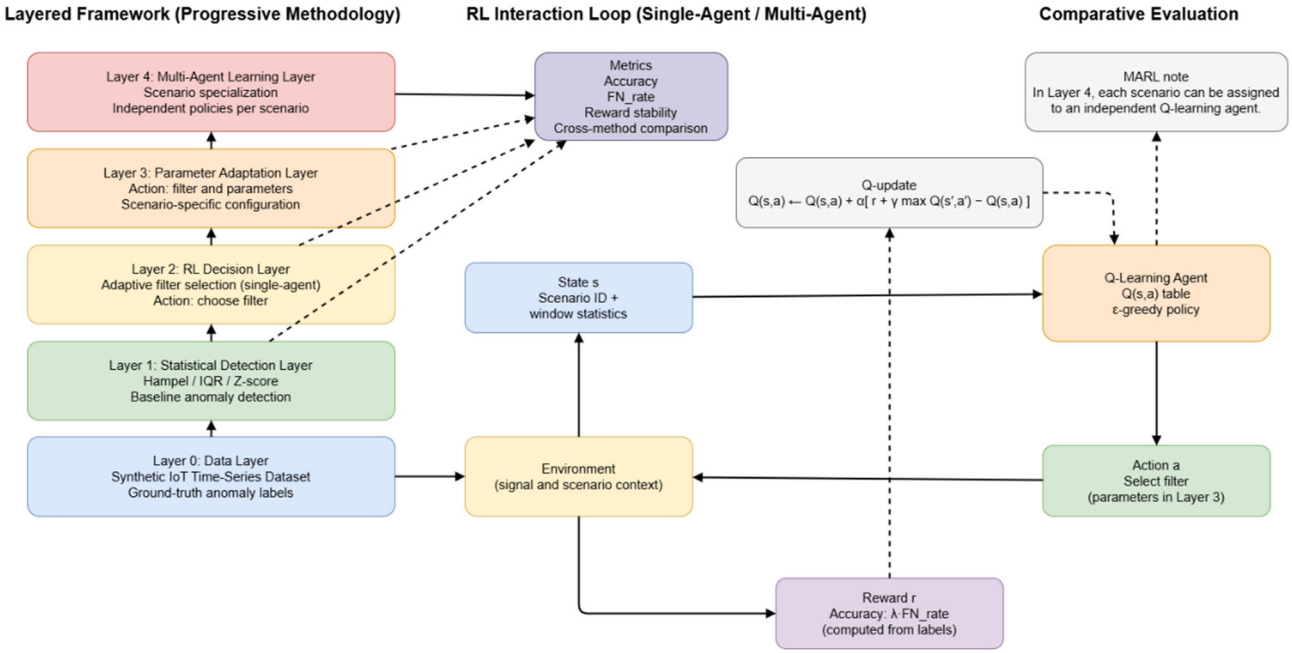

Figure 4. Overall layered framework and RL interaction loop for adaptive anomaly detection in IoT time-series data.

The evaluation component, shown on the right, summarizes the performance assessment using consistent metrics, including detection accuracy, false negative rate, reward stability, and cross-method comparison.

The adaptive selection problem is formulated as a Markov Decision Process (MDP). However, since each episode corresponds to a single-step evaluation of a full dataset, the formulation can also be interpreted as a contextual bandit problem. This formulation can also be interpreted as a contextual bandit problem, given the absence of temporal state transitions and the single-step decision structure. The simplified state representation was intentionally adopted to preserve interpretability and maintain compatibility with lightweight tabular learning in resource-constrained IoT environments.

A compact state representation is adopted to keep the RL design interpretable and aligned with the assessment goals. State (s): scenario identifier, $s \in \{0, 1, 2, 3\}$.

Each state corresponds to one anomaly scenario dataset (stable spikes, impulsive noise, drift/periodic, flat/corrupted). Two action-space designs are considered:

- Filter selection only:

$$a \in \{\text{Hampel}, \text{IQR}, \text{Z-score}\} \tag{1}$$

- Filter and parameter configuration:

$$a \in \{\text{config}_0, \text{config}_1, \ldots, \text{config}_K\} \tag{2}$$

Each action corresponds to a predefined filter–parameter configuration (e.g., Hampel with a window equal to 21 and MAD threshold equal to 3.0), enabling RL to learn both which filter and which parameter set performs best. The reward is designed to reflect two practical priorities:

- High overall correctness (accuracy).
- Strong penalty for missed anomalies (false negatives), which are often more costly in monitoring contexts.

A general reward definition is as follows:

$$r = Accuracy - \alpha \cdot FNR \tag{3}$$

where

- *Accuracy* is computed from true positives (TPs), which denote correctly detected anomalies, true negatives (TNs), which denote correctly identified normal samples, false positives (FPs), which denote normal samples incorrectly classified as anomalies, false negatives (FNs), which denote anomalies incorrectly classified as normal samples over the full scenario, and *FNR*, which represents the false negative rate computed over anomalous samples;
- α controls the severity of false negative penalization.

This reward formulation discourages trivial solutions that achieve high global accuracy while failing to detect anomalous events.

The current system connects the RL state to the dataset scenario, enabling controlled experimental tests. In practical applications, the state representation should incorporate signal statistics such as variance, entropy, and spectral features which combine variance and entropy together with spectral features so that the agent can learn to handle different signal conditions. This limitation is explicitly revisited in the extended stress-test evaluation, where more expressive state representations are introduced to capture inter-node variability and distributed anomaly context while preserving tabular tractability.

3.2. Synthetic Dataset Generation

All scenarios are derived from a common baseline temperature signal, representing a typical IoT environmental sensor operating under nominal conditions. Base signal exhibits smooth temporal evolution, moderate variance, and sampling intervals consistent with low-power IoT monitoring applications. From this baseline, different anomaly mechanisms are injected in a controlled and reproducible manner to simulate realistic degradation patterns. Each generated dataset consists of a univariate temperature time-series accompanied by a binary ground-truth label indicating whether each sample is normal or anomalous.

The anomaly injection process was implemented using reproducible Python scripts with fixed random seeds to ensure controlled and repeatable evaluation conditions. Injected

anomalies included impulsive spikes, gradual drifts, flat corruption segments, periodic perturbations, and correlated multi-node disturbances designed to emulate representative IoT sensing faults and communication disturbances. Ground-truth labels were automatically generated during the injection process and stored together with the corresponding time-series samples.

Signal $temp_t$, which represents temperature, is generated as the sum of three components:

$$temp_t = s_b + n + a \tag{4}$$

where

- s_b represents the baseline environmental temperature signal;
- n represents measurement noise produced by the sensor;
- a represents injected anomalies simulating sensor faults or disturbances.

Baseline temperature signal s_b is modeled as a slowly varying process:

$$s_b = T_0 + \alpha t + \epsilon \tag{5}$$

where

- T_0 is the initial temperature level;
- α represents a small drift coefficient;
- ϵ is low-amplitude Gaussian noise representing natural environmental fluctuations.

Measurement noise component n_t is modeled as follows:

$$n_t \sim \mathcal{N}\left(0, \sigma^2\right) \tag{6}$$

where σ controls the magnitude of sensor noise.

Synthetic anomaly component, a, exhibits different characteristics depending on the defined scenario based on the specific rules determined for each scenario, and demonstrates different levels of intensity and different lengths of time and different chances of occurring throughout its time of operation. Anomaly generation framework uses a stochastic process which scientists can control to create identical testing environments that remain constant throughout all experiments. The proposed architecture enables specific parameters to define each anomaly type by its strength and active time and chance to happen. This design creates spike anomalies through sudden signal changes which produce strong short-term impulse disturbances that break the normal signal pattern.

$$a_{spike} = \begin{cases} A_s & \text{if } t \in \mathcal{T}_{spike} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where A_s represents the spike magnitude and $\mathcal{T}_{spike}$ denotes randomly selected timestamps.

Impulsive noise is modeled as short bursts of high-amplitude deviations applied over small time windows. Gradual drift anomalies are generated as

$$a_{drift} = \beta t \tag{8}$$

where β represents a small drift rate simulating sensor calibration degradation.

Flat corrupted segments are generated by forcing the signal to remain constant during predefined time intervals:

$$x_t = c \tag{9}$$

where c is a constant value representing a frozen sensor output.

The generation process for baseline anomaly scenarios is detailed in Appendix A. The second family of extended synthetic datasets from Appendix B, which was developed to simulate non-stationary overlapping distributed sensing conditions, provides additional support for the stress-test evaluation presented in Section 4.6.

3.3. Statistical Filtering Baseline

Statistical filters form the initial step in the proposed anomaly detection pipeline and serve as a clear benchmark for comparison with learning-based methods. Statistical filters are used as anomaly identification mechanisms rather than purely as signal restoration operators. Their minimal computational demand and the ability to easily understand their functioning make them a viable option for IoT sensing systems that are under energy and processing limitations [10].

In this research, statistical filtering is consciously carried out with fixed parameters, which is typical for real-world deployments where thresholds and window sizes are set manually and seldom changed. This decision allows for an unambiguous evaluation of the restrictions that static detection mechanisms face when different types of anomalies occur simultaneously.

Hampel filter is a robust outlier detection method based on the median and the MAD. Unlike mean-based statistics, the median is insensitive to extreme values, making the Hampel filter well-suited for impulsive noise and isolated spikes. Given the temperature time-series $x = \{x_1, x_2, \ldots, x_N\}$, the Hampel filter operates on a sliding window of size w, which denotes a discrete IoT temperature time-series, where x_i represents the sensor reading at sample index i. For each sample x_i, the local median is computed as follows:

$$\widetilde{x}_i = \text{median}\{x_{i-w}, \ldots, x_{i+w}\} \tag{10}$$

The median absolute deviation (MAD) is defined as follows:

$$\text{MAD}_i = \text{median}\Big(\mid x_j - \widetilde{x}_i \mid\Big), j \in [i-w, i+w] \tag{11}$$

To obtain a consistent estimate of the standard deviation under Gaussian assumptions, the MAD is scaled by a constant factor $c = 1.4826$:

$$\sigma_i = c \cdot \text{MAD}_i \tag{12}$$

A sample is classified as anomalous if:

$$\mid x_i - \widetilde{x}_i \mid > s \cdot \sigma_i \tag{13}$$

where

- w is the window half-size;
- s is the Hampel sensitivity parameter.

Hampel filter is configured with $w = 21$ and $s = 3.0$, consistent with commonly adopted robust detection settings [4]. IQR filter identifies anomalies by analyzing the statistical dispersion of the data through quartile statistics. Let Q_1 and Q_3 denote the first and third quartiles of the time-series within a given window or over the full signal. IQR is defined as follows:

$$\text{IQR} = Q_3 - Q_1 \tag{14}$$

A sample x_i is flagged as anomalous if it lies outside the interval:

$$x_i < Q_1 - k \cdot \text{IQR} \text{or } x_i > Q_3 + k \cdot \text{IQR} \tag{15}$$

where

- k is the IQR scaling factor controlling detection sensitivity.

In this experiment, the value $k = 1.5$ is adopted, following standard conventions in outlier detection and exploratory data analysis [25].

Z-score filter measures how far a sample deviates from the mean of the distribution in units of standard deviation. For a time-series x, the Z-score of a sample x_i is given by the following:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{16}$$

where

- μ is the mean of the series;
- σ is the standard deviation.

A sample is classified as anomalous if:

$$| z_i | > t \tag{17}$$

where

- t is the Z-score threshold.

The threshold is set to $t = 3.0$, corresponding to rare deviations under normal distribution assumptions [26].

Each statistical filter is applied independently to the four scenarios described in the previous section. Detection performance was evaluated using the metrics formally defined in Section 3.7.

3.4. RL for Adaptive Filter Selection

The limitations of fixed statistical filtering methods are discussed in Section 3.3, while Appendix A details the process of synthetic dataset generation and evaluation. The performance of these methods is highly scenario-dependent, varying according to signal characteristics and noise conditions. While individual filters may perform well in specific situations, improved performance can be achieved by adapting filter selection to varying data patterns. This limitation arises from the use of fixed parameters, which impose constant thresholds and window sizes regardless of signal variability. The dynamic IoT environments restrict the effective performance of these methods because their effectiveness decreases under changing system conditions. To address these limitations, RL is introduced as an adaptive method for filter selection. The framework incorporates sequential decision process modeling to learn optimal filtering policies through data interaction, which enables automatic behavior adjustment during actual system operation without requiring manual parameter configuration. The adaptive filter selection problem is modeled as an MDP, defined by the tuple

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \gamma \rangle \tag{18}$$

where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $\mathcal{R}$ the reward function, and γ the discount factor.

Figure 5 presents sequence diagram of a single-agent training episode.

The action space corresponds to the selection of one of the statistical filters introduced in Section 3.4. The formulation follows the definition, previously introduced in Equation (1).

Each action triggers the application of the corresponding filter using fixed parameters defined in the baseline configuration. RL agent learns which filter to apply, while parameter adaptation is addressed in the next extension.

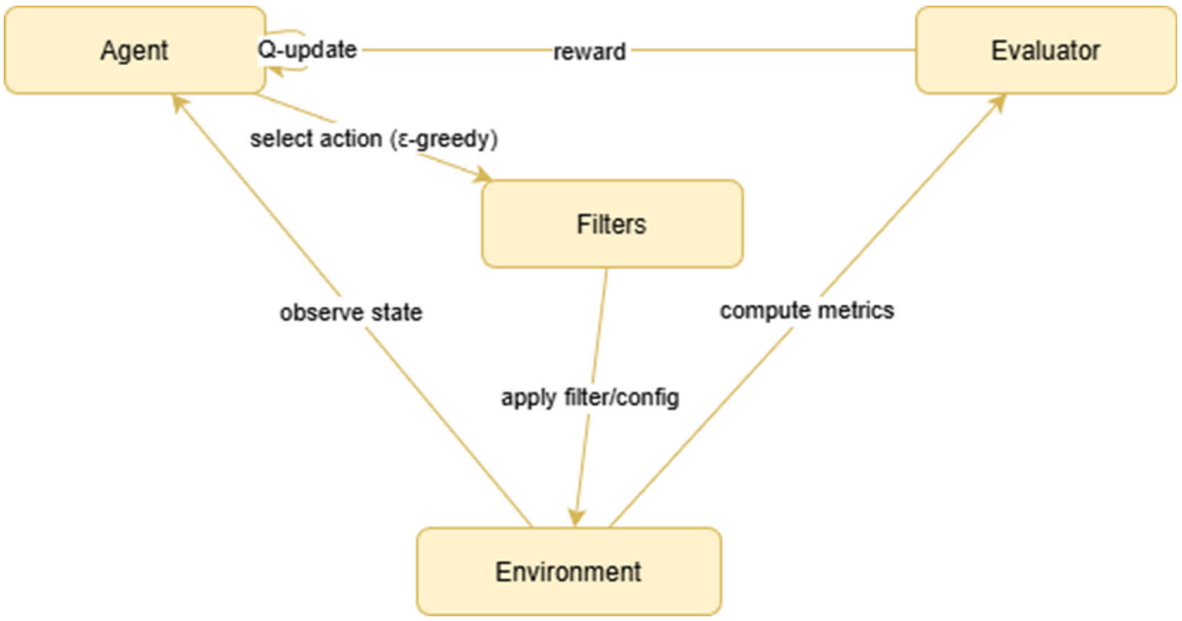

Figure 5. Sequence diagram of a single-agent training episode.

The reward function defined in Equation (2) reflects practical priorities in anomaly detection and is designed to:

- Encourage high overall detection accuracy;
- Penalize missed anomalies (false negatives), which are typically more critical than false positives.

The reward function was designed to balance overall detection performance with strong penalization of missed anomalies, which are particularly critical in IoT monitoring applications.

Single-agent embodiment employs the identical Q-learning strategy presented in Section 3.3 but now directed towards the adaptive choice of statistical filters. The algorithm works on a small state-action space, which allows both stable learning and the full interpretability of the learned policy to coexist. The Q-value update rule is given by the following:

$$Q(s,a) \leftarrow (1-\eta)\, Q(s,a) + \eta \left[r + \gamma \max_{a'} Q(s',a') \right] \tag{19}$$

where

- η is the learning rate;
- γ is the discount factor;
- s' is the next state.

An ε-greedy policy is used to balance exploration and exploitation, with ε decaying over training episodes to favor exploitation as learning converges. Training is performed over a fixed number of episodes. In each episode, a scenario is selected as the current state, and the agent chooses a statistical filter using an ε-greedy policy. The selected filter is applied to the dataset, detection metrics are computed using ground-truth labels, and a reward is calculated. Finally, the Q-table is updated to improve future action selection.

This episodic structure reflects the fact that each scenario evaluation corresponds to a complete filtering decision rather than to a continuous control process.

After convergence, the learned Q-table encodes the agent's preference for each filter under each scenario. Optimal policy is obtained as follows:

$$\pi^*(s) = \arg\max_a Q(s,a) \tag{20}$$

This policy directly maps each scenario to the statistical filter that maximizes the expected reward, providing a transparent and interpretable decision mechanism. Unlike black-box models, the learned policy can be inspected, analyzed, and compared against baseline performance.

Figure 6 illustrates the RL interaction model used in the proposed framework, including the state representation, action space, reward formulation, and Q-learning update mechanism.

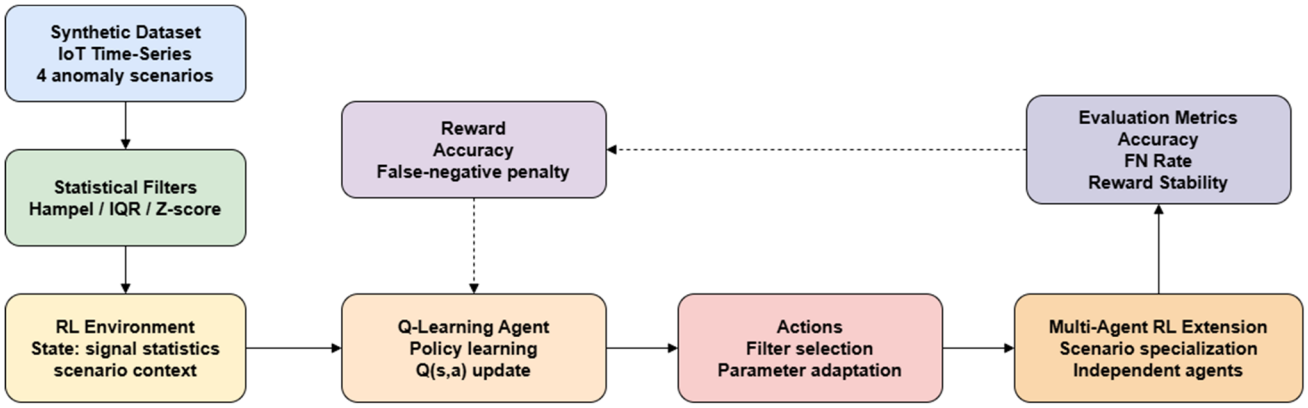

Figure 6. RL architecture used for adaptive anomaly detection.

The agent observes the state derived from IoT sensor data and selects filtering actions. The reward is computed from anomaly detection performance, allowing the agent to learn optimal filtering strategies. The framework is extended to a multi-agent configuration to enable scenario-specific policy learning.

3.5. RL with Parameter Adaptation

In this section, the RL architecture is extended to jointly adapt both the anomaly detection filter and its internal parameters. This enables the agent to automatically identify optimal configurations for each scenario, replacing manual parameter tuning. Extended MDP formulation, state and action definitions, and a training procedure are presented, followed by an interpretation of the learned policies and a discussion of their benefits and limitations.

The single-agent RL method presented in Section 3.4 demonstrated that dynamic filter selection can significantly improve anomaly detection performance compared to static statistical baselines. This method is still dependent on static filter–parameters, which restricts flexibility in the cases where the characteristics of the anomaly change over time.

In real IoT applications, detection performance depends not only on the selected filtering method but also on its parameter configuration. Fixed parameterization always leads to trade-offs among sensitivity, robustness, and detection latency. This section extends the RL framework introduced in the previous section by enabling joint adaptation of both filter selection and parameter configuration. The adaptive filtering problem is again modeled as MDP, in Equation (18). However, both the state representation and the action space are extended compared to the single-agent baseline.

Each state corresponds to one of the four synthetic scenarios defined in Section 3.2. This abstraction ensures that the agent focuses on high-level operating conditions rather than raw signal samples, reducing dimensionality and stabilizing learning.

In the extended formulation, each action corresponds to a specific filter–parameter configuration, rather than to a filter alone. Formally, the action space is defined as follows:

$$\mathcal{A} = \left\{ \left(f, \theta_f \right) \right\} \tag{21}$$

where

- $f \in \{\text{Hampel}, \text{IQR}, \text{Z-score}\}$ denotes the selected filter;
- θ_f represents the parameter set associated with filter f.

The parameter sets used in this experiment are defined as follows:
Hampel filter–parameters:

$$\theta_{\text{Hampel}} = (w, s) \tag{22}$$

where

- w is the sliding window size;
- s is the MAD scaling factor.

IQR filter–parameters:

$$\theta_{\text{IQR}} = (k) \tag{23}$$

where

- k is the IQR multiplier.

Z-score filter–parameters:

$$\theta_{\text{Z}} = (t) \tag{24}$$

where

- t is the Z-score threshold.

A discrete action space is constructed by combining each filter with a small set of candidate parameter values. This design balances expressiveness and tractability, ensuring that the learning problem remains solvable with tabular methods.

The reward function maintains the same structure introduced in Section 3.4, ensuring comparability between Equations (18) and (19).

This formulation encourages the agent to select configurations that maximize detection performance while strongly penalizing missed anomalies. Importantly, the reward is computed after applying the selected filter with its associated parameters, directly linking parameter choices to performance feedback.

However, a single global policy may be insufficient to capture heterogeneous anomaly behaviors across distributed IoT environments, motivating the transition to a multi-agent formulation.

3.6. MARL

This section introduces a MARL extension for dealing with the diversity of IoT anomaly detection cases. System heterogeneity is addressed by assigning different agents to distinct operating conditions. This section describes the multi-agent architecture, the learning process of individual agents, and the analysis of the resulting policies [8,27–31].

The RL formulations presented in Sections 3.4 and 3.5 rely on a single decision-making agent responsible for selecting the optimal filter or filter–parameter configuration across all operating conditions. While effective, this approach implicitly assumes that a single global policy can adequately capture the diversity of anomaly behaviors encountered in heterogeneous IoT environments.

In real-world deployments, different sensing nodes or operating contexts may experience distinct disturbance profiles, making decentralized or specialized learning strategies

more appropriate. MARL provides a natural extension by enabling multiple agents to learn concurrently, each focusing on a specific operating condition while collectively contributing to improved system robustness.

The proposed MARL framework allows agents to cooperate implicitly through shared evaluation metrics while learning scenario-specialized policies for adaptive anomaly detection.

The proposed MARL framework consists of a set of agents:

$$\mathcal{A} = \{A_0, A_1, A_2, A_3\} \tag{25}$$

where each agent A_i is associated with one anomaly scenario defined in Section 3. This design enables scenario-level specialization, allowing each agent to focus exclusively on the anomaly characteristics of its assigned environment.

All agents share:

- The same set of statistical filters.
- The same parameter configuration space.
- The same reward definition.

However, each agent maintains its own Q-table and learns independently from its own interaction history. For each agent A_i, the state space is defined as follows: $s_i = i$. Each agent operates in a fixed state corresponding to its associated scenario. This simplified formulation avoids unnecessary state transitions and ensures stable learning.

The action space for each agent is identical to the one defined in the previous section:

$$a \in \left\{ \left(f, \theta_f \right) \right\} \tag{26}$$

where f denotes the selected filter and θ_f is its parameter configuration. This allows direct comparison between single-agent and multi-agent approaches.

Each agent receives a reward computed locally based on its scenario-specific detection performance:

$$r_i = \text{Accuracy}_i - \alpha \cdot \text{FN\_rate}_i \tag{27}$$

The baseline formulation avoids inter-agent communication, maintaining low computational overhead suitable for resource-constrained IoT environments. The system has been designed this way because organizations which operate resource-limited IoT activities need to control their resources through coordination processes which would need extra computing power and time for system alignment and costs for data exchange. The test determines whether implicit specialization and policy diversity but reward shaping with its current low-execution costs will deliver satisfactory results. This design enables multi-agent operation without direct communication, making it suitable for edge and embedded IoT systems. The present study does not need advanced coordination methods which authors can investigate in their future studies.

3.7. Evaluation Metrics

Due to the class imbalance commonly observed in anomaly detection tasks, performance evaluation was primarily based on recall, F1-score, false positive rate (FPR), and false negative rate (FNR), while accuracy was used only as a complementary global indicator. Detection performance was assessed by comparing predicted anomalies with ground-truth labels using the following metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{28}$$

$$Precision = \frac{TP}{TP + FP} \tag{29}$$

$$Recall = \frac{TP}{TP + FN} \tag{30}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{31}$$

$$FPR = \frac{FP}{FP + TN} \tag{32}$$

$$FNR = \frac{FN}{TP + FN} \tag{33}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. These metrics were directly computed in the Python 3.11 implementation [32] and used for both baseline comparison and RL reward evaluation throughout the experimental analysis.

4. Results

4.1. Dataset and Experimental Setup Overview

A synthetic data approach is adopted to enable controlled, reproducible, and well-characterized anomaly patterns representative of real IoT sensing conditions. This section describes the base signal model, the definition of noise and fault scenarios, and the ground-truth labeling process, and provides a summary of the resulting datasets and their availability for reproducibility. Table 3 summarizes the main characteristics of the four generated scenarios, including anomaly types and associated detection challenges.

Table 3. Summary of synthetic IoT anomaly scenarios.

Scenario	Anomaly Type	Temporal Structure	Main Detection Challenge
1	Sporadic spikes	Isolated events	Avoid false positives
2	Impulsive noise	Dense, irregular	Robustness to noise
3	Drift and periodic	Progressive patterns	Contextual detection
4	Flat-line/corruption	Persistent failure	Detect low-variance faults

All datasets were generated using reproducible Python scripts with fixed random seeds and exported as CSV files to ensure transparency and repeatability. The anomaly injection process was fully controlled and automatically labeled during dataset generation, enabling consistent evaluation across all experimental scenarios. The complete experimental framework, including dataset generation scripts, RL and MARL implementations, benchmark evaluation utilities, and plotting tools, is publicly available in the repository referenced in [33]. Four experimental approaches are evaluated, as illustrated in Figure 7, and compared in terms of decision mechanism, adaptivity, and learning strategy.

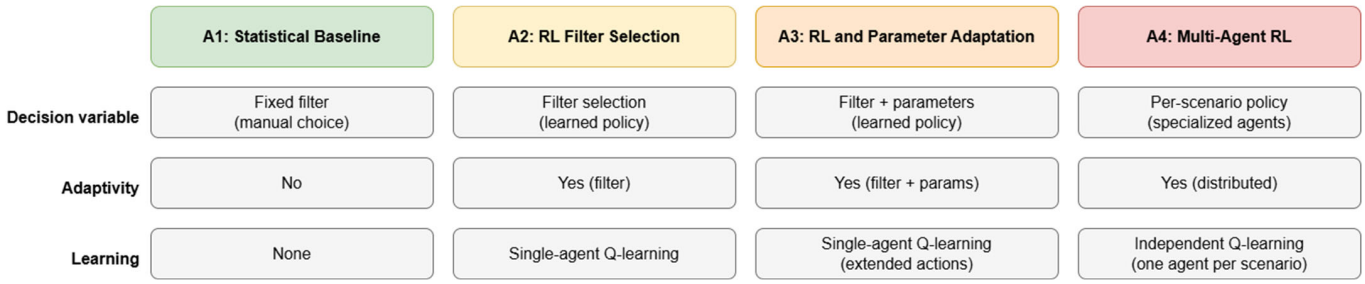

Figure 7. Comparison of the four evaluated approaches in terms of adaptivity and decision mechanism.

The first approach (A1) corresponds to the statistical baseline which uses fixed filters (Hampel, IQR, or Z-score) to detect anomalies through manually defined parameters. The filtering method operates without any learning system because the filtering approach maintains its unchanging state. The second approach (A2) uses RL to create a system which automatically selects the appropriate filters. A single Q-learning agent learns to choose the most suitable filter according to the operating scenario, enabling adaptive behavior while keeping filter–parameters fixed. The third approach (A3) extends the action space of the RL agent by allowing joint optimization of the filter and its parameters. The learning process enables the development of scenario-specific configurations which provide greater operational flexibility than basic filter selection methods. The fourth approach (A4) presents a MARL system which assigns Q-learning agents to work on different anomaly detection tasks. The distributed learning approach enables agents to focus on operational conditions while they share the same set of actions. Figure 7 demonstrates how anomaly detection methods in IoT time-series evolved from using fixed statistical techniques to adopting learning methods which use adaptive and distributed processing.

4.2. Performance of Statistical Filtering Baseline

Figure 8 illustrates the behavior of the Hampel, IQR, and Z-score filters under Scenario 2 (impulsive noise). The figure highlights the contrasting responses of median-based, quartile-based, and mean-based methods to dense impulsive disturbances.

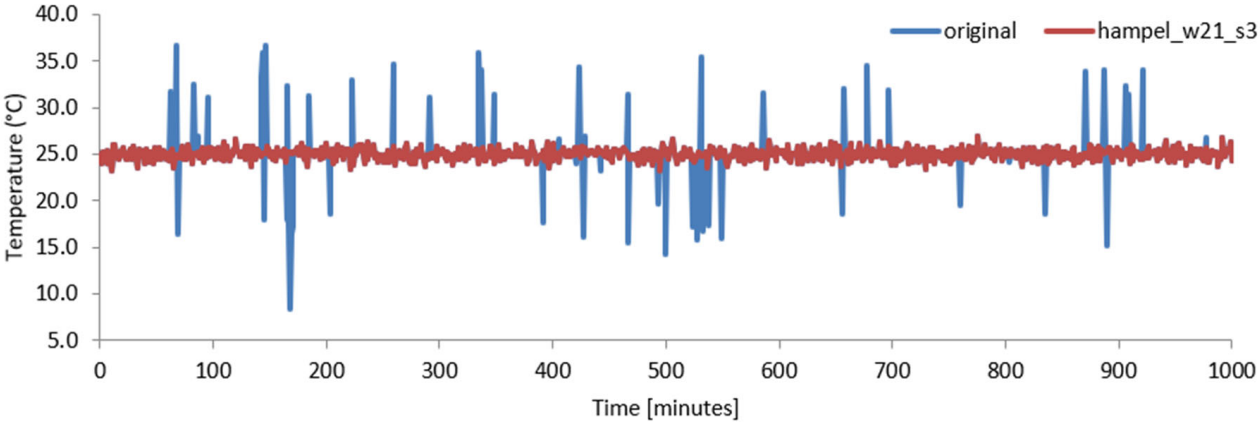

Figure 8. Scenario 2: original signal versus Hampel filter output ($w = 21$, $s = 3.0$).

The Hampel filter effectively suppresses isolated outliers while preserving the underlying signal trend, demonstrating robustness to impulsive noise. IQR filter demonstrates higher sensitivity, occasionally affecting normal samples. Z-score filters exhibit instability due to their reliance on global mean and variance, which are influenced by extreme values.

To complement the quantitative metrics presented earlier, a qualitative visual analysis was performed to illustrate the behavior of the statistical filters under impulsive noise conditions. Scenario 2 was selected as a representative and challenging case, as it contains frequent and abrupt outliers that stress the robustness of fixed-parameter detection methods. Figures 8–10 compare the original temperature time-series with the outputs of the Hampel, IQR, and Z-score filters using the parameter configurations defined in previous sections.

The Hampel filter effectively suppresses impulsive outliers while preserving the underlying signal structure, demonstrating its robustness to isolated extreme values.

The IQR filter exhibits higher sensitivity to impulsive noise, occasionally affecting normal samples due to its reliance on global quartile statistics.

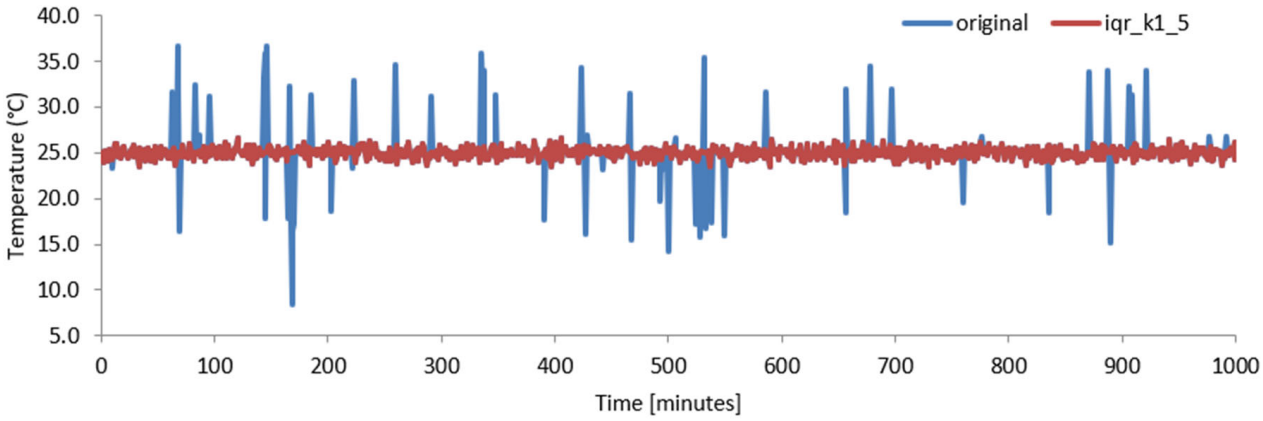

Figure 9. Scenario 2: original signal versus IQR filter output ($k = 1.5$).

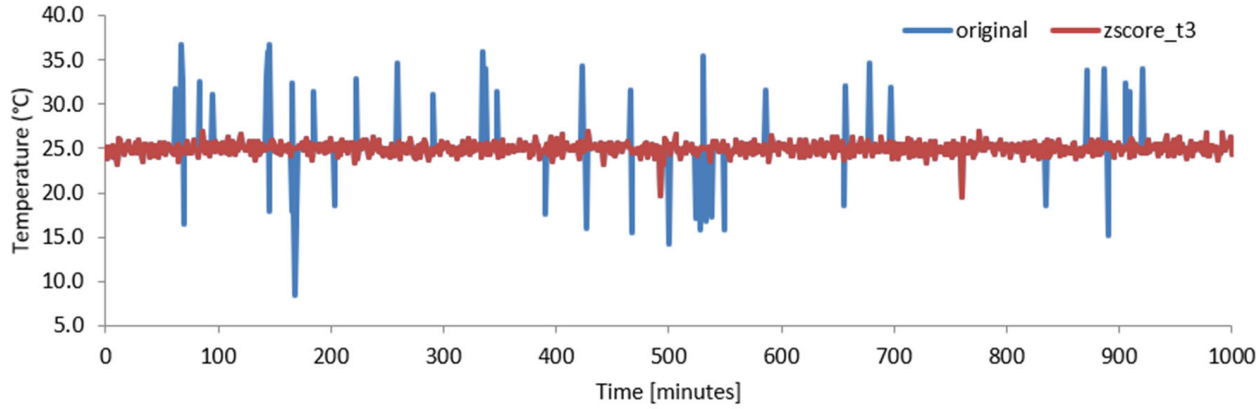

Figure 10. Scenario 2: original signal versus Z-score filter output ($t = 3.0$).

The Z-score filter shows instability under impulsive noise conditions, as mean and variance estimates are influenced by extreme values, leading to less reliable anomaly suppression.

Qualitative comparison confirms that the performance of statistical filters is strongly scenario-dependent. While the Hampel filter performs well under impulsive noise, no fixed configuration provides consistently robust behavior across all anomaly types. This qualitative evidence reinforces the limitations of static filtering and motivates adaptive reinforcement learning strategies.

4.3. RL with Filter–Parameter Adaptation

The learned policy (Equation (29)) maps each anomaly scenario to an optimal filter–parameter configuration, providing a transparent and interpretable decision rule. This interpretability is particularly valuable in IoT contexts, where understanding why a given configuration is selected is often as important as raw performance.

Empirical results indicate that the agent tends favor:

- Robust median-based configurations under impulsive noise;
- More sensitive parameterizations in stable environments;
- Conservative thresholds in severely corrupted signals.

4.4. MARL Performance

Each agent independently applies the previously defined Q-learning update rule, keeping their own Q-table and exploration policy. This independent learning system allows

the agents to become experts in the scenario while the simplicity and interpretability of tabular reinforcement learning are maintained.

After training convergence, each agent yields a scenario-specific optimal policy:

$$\pi_i^* = \arg\max_a Q_i(s_i, a) \tag{34}$$

This results in policy specialization, where

- Agents operating under impulsive noise favor robust median-based configurations;
- Agents handling stable signals prefer sensitive configurations;
- Agents exposed to severe corruption adopt conservative parameter choices.

The MARL formulation improves training organizations by enabling scenario-specific policy learning. The main experiments evaluated controlled scenarios which showed this method improves training organization and system interpretability although it does not consistently improve detection accuracy.

The multi-agent framework offers several advantages over the single-agent formulation:

- Improved specialization: each agent optimizes for a specific anomaly profile.
- Faster convergence: reduced state-action ambiguity accelerates learning.
- Scalability: new agents can be added for additional scenarios without retraining the entire system.

However, MARL introduces additional training overhead and requires careful coordination in distributed deployments.

Learning dynamics of the multi-agent system are illustrated in Figure 11, which shows the evolution of the reward over training episodes for each agent. The results indicate stable learning behavior across all agents, with convergence towards high reward values and occasional drops associated with ε-greedy exploration and scenario-specific uncertainty. Importantly, no signs of divergence or instability are observed, confirming the robustness of the learning process.

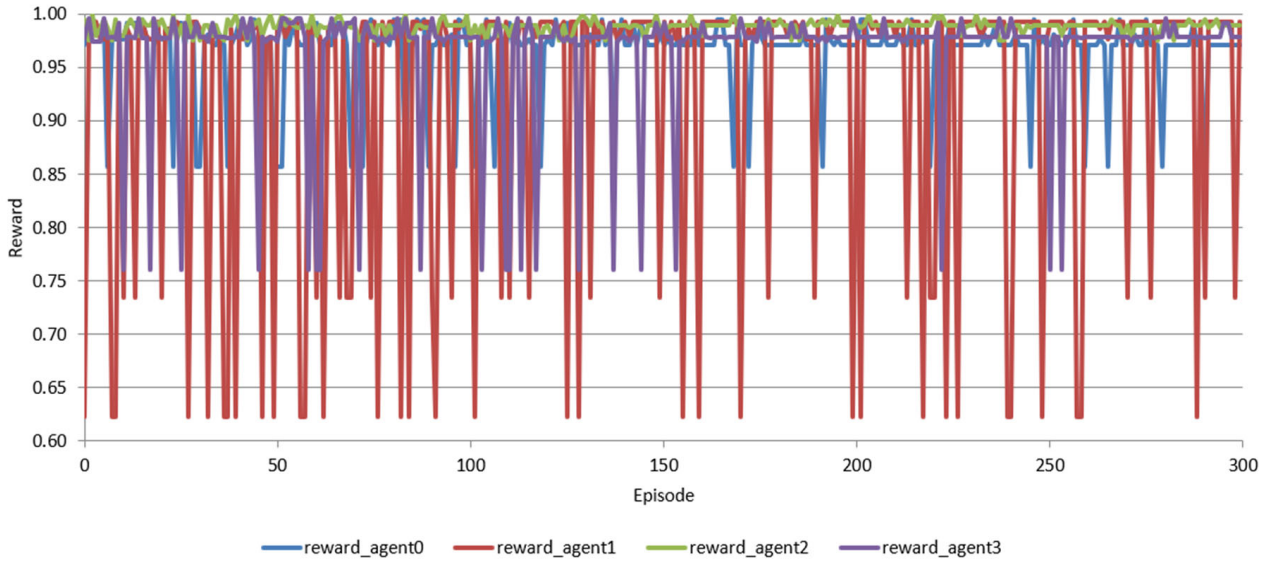

Figure 11. Training reward evolution per agent.

The final decision outcomes can be seen in Table 4. It shows the optimal filter–parameter configuration that has been learned for each scenario by the single-agent and multi-agent approaches. From the table, it is evident that both configurations arrive at the same optimal solutions, thus confirming the consistency and accuracy of the policies learned. However,

the multi-agent formulation still accommodates scenario-level decoupling during training which minimizes policy interference and consequently improves learning stability.

Table 4. Optimal filter–parameter configuration per scenario.

Scenario	Single-Agent Optimal Configuration	Multi-Agent Optimal Configuration
0	IQR (w = 21, k = 1.5)	IQR (w = 21, k = 1.5)
1	IQR (w = 31, k = 2.0)	IQR (w = 31, k = 2.0)
2	Z-Score (w = 21, t = 3.0)	Z-Score (w = 21, t = 3.0)
3	Z-Score (w = 31, t = 2.5)	Z-Score (w = 31, t = 2.5)

The first one is a clear and reproducible experimental validation of the MARL strategy, while the latter one captures the final decision logic. This fusion, together with Figure 11 and Table 3, gives an insight into the proposed approach that is understandable and reproducible. Firstly, it emphasizes learning dynamics, whereas the second one reflects final decision logic.

4.5. Cross-Method Comparative Analysis

Statistical filters such as Hampel, IQR, and Z-Score serve as the baseline for the proposed framework. While these methods perform well in stable scenarios, their effectiveness decreases under drift, corruption, or rapidly changing conditions, where fixed thresholds often increase false negative or delay detection. To improve adaptability, the RL-based approach dynamically selects the most suitable filter according to the observed scenario, reducing the need for manual tuning and improving robustness across heterogeneous conditions. Extending the action space to include parameter adaptation further improves stability and reduces detection errors, showing that parameter tuning is as important as filter selection itself. The MARL formulation further improves convergence stability by assigning specialized agents to different scenarios, reducing policy interference and enabling more modular learning behavior. The final policies obtained by the single-agent and multi-agent approaches are summarized in Table 4, while Table 5 and Figure 12 provide a comparative overview of performance across scenarios. Although static statistical filters can still achieve strong results when the optimal configuration is known beforehand, RL and MARL provide greater flexibility for realistic IoT environments where operating conditions evolve over time.

All methods tested in the study demonstrated total accuracy because they reported no false negative results across all test scenarios. Detection of all injected anomalies succeeded because the synthetic dataset provided controlled conditions which included clear definitions of anomalies that could be statistically distinguished from normal patterns. A comparison of methods shows different results because methods generate different rates of false positive errors and they show varying levels of detection consistency instead of showing true detection errors. The method which achieves best results among all tested methods uses a static configuration with scenario-specific tuning to obtain its best accuracy rating of 0.996 because it combines existing knowledge with complete control over parameter settings. The static single-configuration approach also demonstrates strong performance (0.994) because basic statistical filters can maintain their effectiveness across various situations without requiring adjustments. Single-agent and multi-agent reinforcement learning methods reached an average accuracy of 0.990, which represents a minor decrease from their achieved results. Learning agents require different scenarios of handling competencies, which result in suboptimal performance outcomes that fail to meet their complete potential. The multi-agent method establishes dedicated scenarios for its operations, yet the artificial anomalies present restricted acting possibilities which

reduce their potential advantage from dedicated scenarios. Near-zero FN values were expected in several synthetic scenarios due to the controlled anomaly injection process and the relatively distinct statistical deviation patterns.

Table 5. Quantitative comparison of accuracy, FNR, and reward across scenarios for static baselines and learning-based approaches.

Scenario	Method	Accuracy	FNR	Reward
0	Static (single config): IQR (w = 31, k = 2.0)	0.995	0.000	0.995
1	Static (single config): IQR (w = 31, k = 2.0)	0.993	0.000	0.993
2	Static (single config): IQR (w = 31, k = 2.0)	0.995	0.000	0.995
3	Static (single config): IQR (w = 31, k = 2.0)	0.992	0.000	0.992
0	Static (best per scenario)	0.995	0.000	0.995
1	Static (best per scenario)	0.993	0.000	0.993
2	Static (best per scenario)	1.000	0.000	1.000
3	Static (best per scenario)	0.996	0.000	0.996
0	RL single-agent (filter and param)	0.971	0.000	0.971
1	RL single-agent (filter and param)	0.993	0.000	0.993
2	RL single-agent (filter and param)	1.000	0.000	1.000
3	RL single-agent (filter and param)	0.996	0.000	0.996
0	MARL (one agent per scenario)	0.971	0.000	0.971
1	MARL (one agent per scenario)	0.993	0.000	0.993
2	MARL (one agent per scenario)	1.000	0.000	1.000
3	MARL (one agent per scenario)	0.996	0.000	0.996
Avg	Static (single config): IQR (w = 31, k = 2.0)	0.994	0.000	0.994
Avg	Static (best per scenario)	0.996	0.000	0.996
Avg	RL single-agent (filter and param)	0.990	0.000	0.990
Avg	MARL (one agent per scenario)	0.990	0.000	0.990

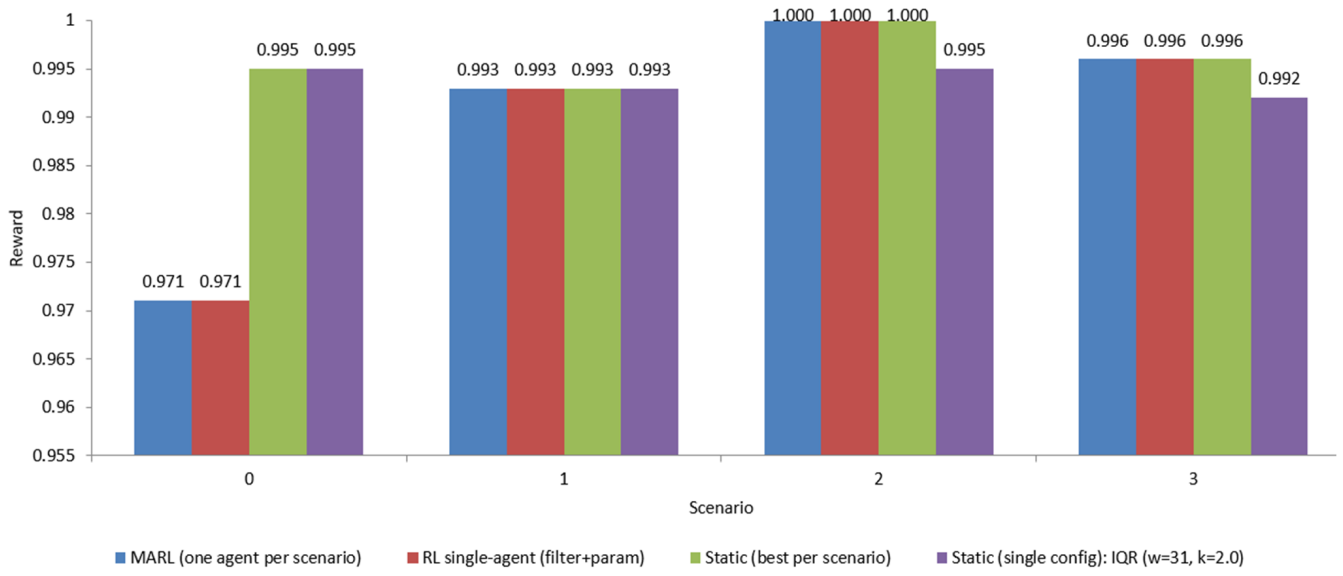

Figure 12. Reward comparison by scenario and method (static baselines vs. RL and MARL).

Figure 12 presents a graphical representation that shows reward values for every scenario and all tested methods, which serves as an additional result to the findings that

Table 4 presents. The methods achieve their highest results because all of them maintain performance levels above 0.97 throughout all tested scenarios.

The static method with scenario-specific tuning achieves the highest performance across most scenarios, confirming that manually optimized statistical configurations can approach optimal detection behavior in controlled environments. Both single-agent and multi-agent RL approaches demonstrate stable performance which maintains consistency but operates at lower levels than the best static configuration. Learned policies developed in Scenario 0 show a performance gap that results from their decision to select less aggressive configurations, which increases false positive rates while maintaining zero false negatives.

Single-agent and multi-agent systems show identical performance results because scenario-based specialization fails to deliver any additional advantages in the tested conditions. The system exhibits this behavior because the anomaly patterns show low complexity and high separability, which enables a single policy to capture the fundamental structure of the patterns.

Research findings demonstrate that RL enables adaptive learning capabilities, but its benefits are restricted to basic well-organized environments.

The results presented so far indicate that, under controlled and highly separable synthetic conditions, the single-agent and multi-agent formulations converge to nearly equivalent policies. The study shows that MARL advantages will remain hidden until the anomaly detection problem starts to exhibit different time patterns and different structural patterns. The next subsection introduces an extra stress-test evaluation which uses more advanced synthetic datasets that are explained in Appendix B.

4.6. Stress-Test Evaluation for MARL Under Complex and Distributed Conditions

Three additional synthetic scenarios were designed to evaluate system performance under more complex conditions.

Scenario 5 models regime switching behavior, where the signal alternates between different statistical properties throughout time. Scenario 6 introduces overlapping anomaly patterns, which combine multiple anomaly types at the same time, making detection more difficult to perform. Scenario 7 simulates multi-node correlated anomalies, which occur when multiple sensors show synchronized abnormal behavior that mirrors distributed Internet of Things systems.

The MARL system demonstrates competitive abilities in Scenario 7 while achieving better results than the single-agent method. The system demonstrates that distributed learning approaches achieve better results because distributed learning enables spatial pattern detection through multiple nodes, which allows agents to develop specialized skills without needing to work together. Dataset descriptions in Appendix B provide a foundation to reproduce results while conducting further assessments of the dataset. The primary advantage of the MARL formulation lies not in immediate performance improvement, but in scalability, modularity, and the ability to handle distributed and heterogeneous IoT environments.

Results indicate that MARL does not deliver performance boosts across all situations. However, it shows advantages in structured environments with distributed and correlated anomalies. The system maintains low processing demands because agents do not need to share information through direct communication, which makes it appropriate for use in IoT devices with limited resources.

Table 6 results show two different performance patterns which separate single-agent RL from MARL methods. The single-agent RL system achieves better results in Scenario 5 because its centralized decision-making system works better when facing sudden changes that happen in short time periods. Two approaches in Scenario 6 produced lower per-

formance results because the testing environment became more complicated through overlapping anomaly patterns, which resulted in higher false negative rates and lower detection success rates.

Table 6. Quantitative comparison on the stress-test datasets for single-agent RL and MARL.

Stress-Test Scenario	Method	Accuracy	FN Rate	Reward
Scenario 5–Regime switching	RL single-agent (filter param.)	0.5222	0.5724	0.2073
Scenario 5–Regime switching	MARL	0.4831	0.5746	0.1654
Scenario 6–Overlapping anomalies	RL single-agent (filter param.)	0.2127	0.8340	−0.3404
Scenario 6–Overlapping anomalies	MARL	0.1897	0.8326	−0.3601
Scenario 7–Multi-node correlated	RL single-agent (filter param.)	0.8664	0.5178	0.5319
Scenario 7–Multi-node correlated	MARL (improved)	0.8580	0.5361	0.5526
Average	RL single-agent (filter param.)	0.5338	0.6414	0.1329
Average	MARL/MARL (improved in Scenario 7)	0.5103	0.6478	0.1193

The authors design new stress-test datasets to assess evaluation situations (Table 6) which would show how multi-agent learning provides advantages. New datasets introduce three elements which create operational differences compared to the original scenarios through their ability to control time-based changes and their capacity to simulate multiple anomaly detection systems and their capability to show node interaction. The main limitation of the previous MARL system was its original dataset, which contained insufficient structural elements needed to develop specific scenarios for assessment purposes. This concern is already consistent with the manuscript's own discussion, which acknowledges that the original synthetic scenarios are sufficiently separable for both single-agent and multi-agent learning to converge to the same effective policies.

Table 6 presents three different types of results. Scenario 5 shows that single-agent RL outperforms MARL through all three performance indicators. The current system shows that temporal regime switching needs more than its current capacity to deliver major benefits through agent decomposition. Two methods under study in Scenario 6 demonstrate their expected poor results because anomaly types present high levels of uncertainty and overlap with each other. Negative reward values in this scenario are not an error; they exist because of how the reward function operates, which uses false negative penalties as its main component when the detector misses an essential number of anomalous samples. The advanced MARL system in Scenario 7 produces better rewards than the single-agent system, although the latter maintains a small advantage in accuracy and false negative detection. Modified MARL formulation shows its ability to use problem distribution, but this advantage operates better through policy coordination than it does through traditional detection methods.

The stress-test analysis indicates that MARL does not universally outperform single-agent RL. Research demonstrates that MARL increases its importance for anomaly detection when the system shows distributed patterns and different structural components and partial system visibility, but this development depends on the implementation of state and reward and decision fusion methods. The position requires more evidence

because it makes a claim of superiority, which does not match the evidence which the experiments provided.

Figure 13 shows a study that compares single-agent RL with MARL, to assess their performance through various stress-test scenarios. Results indicate that different scenarios produce distinct performance outcomes, which result in no single method of achieving consistent success. MARL exhibits a competitive performance in the multi-node correlated scenario, but its advantages become limited, which shows the need for proper coordination design and reward formulation methods in multi-agent environments.

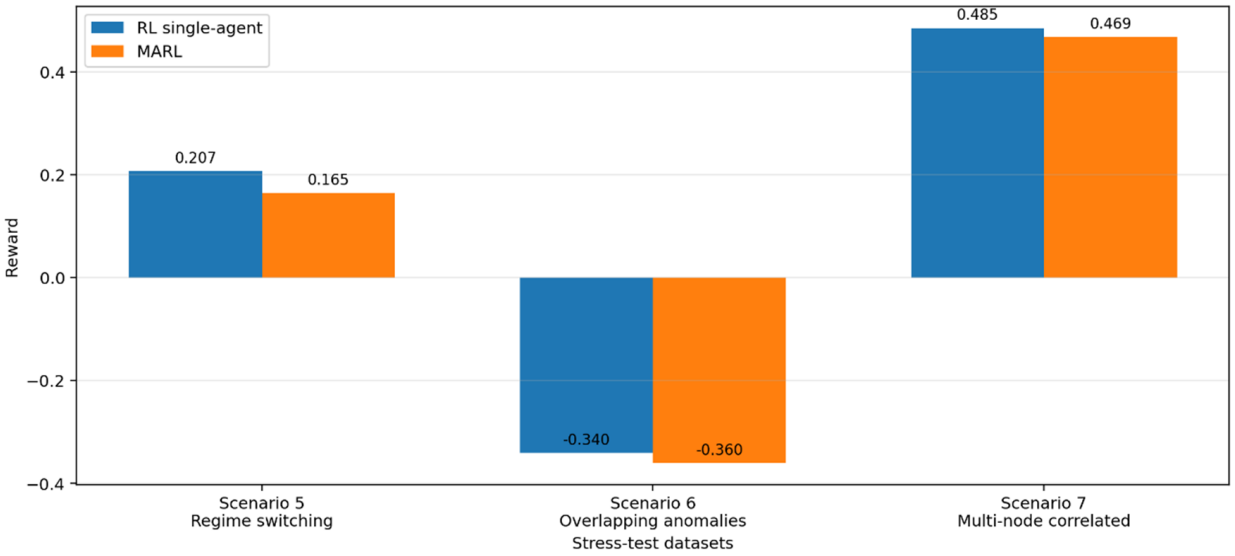

Figure 13. Reward comparison between single-agent RL and MARL.

Reward comparison between single-agent RL and MARL methods shows their performance in three extended stress-test scenarios through Figure 13. Evaluated scenarios appear on the horizontal axis, while the achieved reward values show up on the vertical axis. The single-agent RL system demonstrates superior performance to MARL in Scenario 5 because centralized decision-making handles sudden time changes and non-stationary signal patterns more effectively. In Scenario 6 (overlapping anomalies), both approaches show performance decline because they produce negative reward results. The increased confusion about anomaly detection patterns results in higher false negative detection rates which make it harder to identify anomalies. The MARL system shows competitive performance results which exceed single-agent performance during Scenario 7 because of its ability to handle multiple-node correlated anomalies. The results indicate that distributed learning helps systems to manage populated areas better because local agent expertise improves overall system performance. Figure 13 shows that MARL fails to deliver performance improvements in every situation because it only produces benefits in environments with defined structures which enable localized system changes based on inter-node connections.

Initially, the MARL framework was operated through a training method which divided scenarios into separate parts, allowing each agent to focus on one scenario while maintaining tabular Q-learning simplicity and interpretability. While the experimental evaluation confirms the correct operation of the framework, the current MARL implementation does not include explicit distributed communication between agents. The study indicates that the main advantages of the system derive from its ability to create specialized units that reduce policy conflicts while allowing separate system components to function independently, which results in better decision-making in environments with linked sensing systems.

The restriction becomes apparent when analyzing the stress-test datasets. The anomaly structure in Scenarios 5 and 6 maintains its local nature, despite the increased temporal complexity. The system does not obtain significant advantages from distributing tasks to different agents. The complete understanding of anomalies in Scenario 7 requires agents to analyze relationships between different nodes, which makes original local-only MARL design insufficient because agents depend mostly on their individual signals. The enhanced MARL system changes three fundamental aspects of the learning process through changes to its state representation, reward system, and decision-making process.

The original reward definition in this work was based on the practical anomaly detection priorities of high correctness and strong false negative penalization, and was expressed in the generic form, present in Equation (2), which is fully consistent with the methodological foundations of the article. Stress-test extension maintains the original concept but extends it to demonstrate how the problem functions across multiple distributed systems. The local state of each agent i received extra information through inter-node statistics:

$$s_i = \left[\sigma_i^2, \mid \beta_i \mid, J_i, D_i, \sigma_{\text{global}}, \Delta_i\right] \tag{35}$$

Here, σ_i^2 denotes the local variance of node i, $\mid \beta_i \mid$ is the absolute local slope estimated over the current window, J_i is a jump-count indicator derived from abrupt temporal differences, D_i is the average discrepancy between the local signal and the mean of the neighboring nodes, σ_{global} is the global standard deviation across all nodes, and Δ_i is the absolute difference between the local mean and the global mean. These terms were discretized into a tabular state so that the overall MARL model remained compatible with the article's low-complexity and interpretable Q-learning framework.

This modification is important because it gives each agent access to its own signal behavior. The system provides agents with a compact representation that shows whether the observed deviation will stay local or spread across the entire system. These requirements are important in distributed IoT sensing because it needs to handle a single corrupted node according to different rules than the actual event, which affects multiple nodes at once. The local reward of each agent was defined as

$$R_i^{\text{local}} = \text{Accuracy}_i - 0.7 \cdot \text{FN}_{rate,i} + 0.2 \cdot F1_i \tag{36}$$

The new reward system maintains the same focus on missed anomalies which existed in the original system but introduces the F1 term to enhance performance assessment when testing difficult scenarios. The inclusion of F1 is justified here because Scenario 7 no longer represents a purely local binary decision problem; it includes global events and local faults simultaneously, making harmonic balance between detection sensitivity and error control more relevant.

To reflect the system-wide objective, a global reward term was introduced:

$$R^{\text{global}} = \frac{1}{N}\sum_{i=1}^{N} R_i^{\text{local}} \tag{37}$$

where N is the number of sensing nodes. This term summarizes the average utility of the collective behavior across the distributed environment.

To encourage coherent decisions across agents, a consistency term was also defined:

$$C = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(a_i = a_{\text{majority}}\right) \tag{38}$$

where a_i is the local action which is selected by agent i, a_{majority} is the most frequently selected action in the current episode, and $\mathbb{1}(\cdot)$ is the indicator function. This term takes

values in $[0,1]$, with higher values indicating stronger agreement among agents. The purpose of this score is not to force identical behavior in every case, but to discourage unstable divergence when the observations suggest a globally correlated event. In practice, it acts as a regularize on the policy space.

The final reward used to update each agent was then defined as

$$R_i = \lambda_g R^{\text{global}} + \lambda_l R_i^{\text{local}} + \lambda_c C \tag{39}$$

with

$$\lambda_g = 0.6, \lambda_l = 0.2, \lambda_c = 0.2.$$

These weights were chosen to make the global objective dominant while still preserving local specialization and coordinated consistency. The rationale is straightforward: in Scenario 7, the key challenge is no longer simply to optimize a local detector, but to learn a policy that behaves sensibly in the presence of correlated sensing nodes. Finally, the local actions were aggregated using a majority-voting rule:

$$a_{\text{final}} = \text{mode}(a_1, a_2, \ldots, a_N) \tag{40}$$

The final system decision occurs through complete cooperative teamwork. The improved MARL system achieved better results in Scenario 7 because it uses better methods, which lead to better outcomes yet fall short of the single-agent baseline, which measures accuracy and false negative rate. Stress-test evaluation strengthens the manuscript in two ways. The first test shows that the original MARL from the main experiments failed to deliver advantages because the original synthetic scenario-controlled tests were highly separated in their design. Second, it demonstrates that, once the problem is made more distributed and structurally heterogeneous, MARL begins to show measurable benefits, although that benefit currently emerges more clearly in reward than in raw detection metrics.

Scientific research shows better outcomes when direct proof of superiority exists. The proposed framework demonstrates its actual limits through its demonstrated capabilities. The following conclusion states that single-agent RL works for simple and moderately complex local anomaly patterns. However, MARL demonstrates clear advantages in scenarios involving distributed and correlated anomalies, where decentralized learning improves system-level awareness. This behavior is particularly evident in scenarios involving correlated and distributed anomalies, where MARL demonstrates clear advantages over centralized learning.

4.7. Real-World Evaluation with RL and MARL

A second evaluation was conducted using real-world IoT temperature data to test their adaptive filtering framework by using actual IoT temperature data which they collected from various types of sensor devices. The experiment tests the proposed approach by using actual acquisition patterns and sensor devices that demonstrate different sampling rates and performance characteristics.

A real-world dataset was obtained from the public repository [34], comprising temperature measurements collected from multiple distributed IoT sensor nodes. After preprocessing, which included data cleaning and the removal of incomplete or corrupt records, nine sensors were retained for analysis. Each sensor is modeled as an independent univariate time-series, preserving its original temporal structure. In contrast to synthetic datasets, the real data exhibits significant heterogeneity across sensors, including variations in signal amplitude, temporal continuity, noise levels, and anomaly manifestation patterns. These characteristics reflect realistic operating conditions, where sensor behavior is influenced by environmental factors and system-level variability. Such diversity

provides a more challenging and representative testbed for evaluating adaptive filtering strategies. Figure 14 illustrates a representative example of a real sensor signal with injected anomalies. The figure highlights both the inherent variability of the temperature signal and the diversity of anomaly patterns embedded in the data, including abrupt deviations, gradual shifts, and irregular disturbances. This dataset is therefore used to assess the robustness and generalization capability of the proposed adaptive filtering framework under real-world conditions.

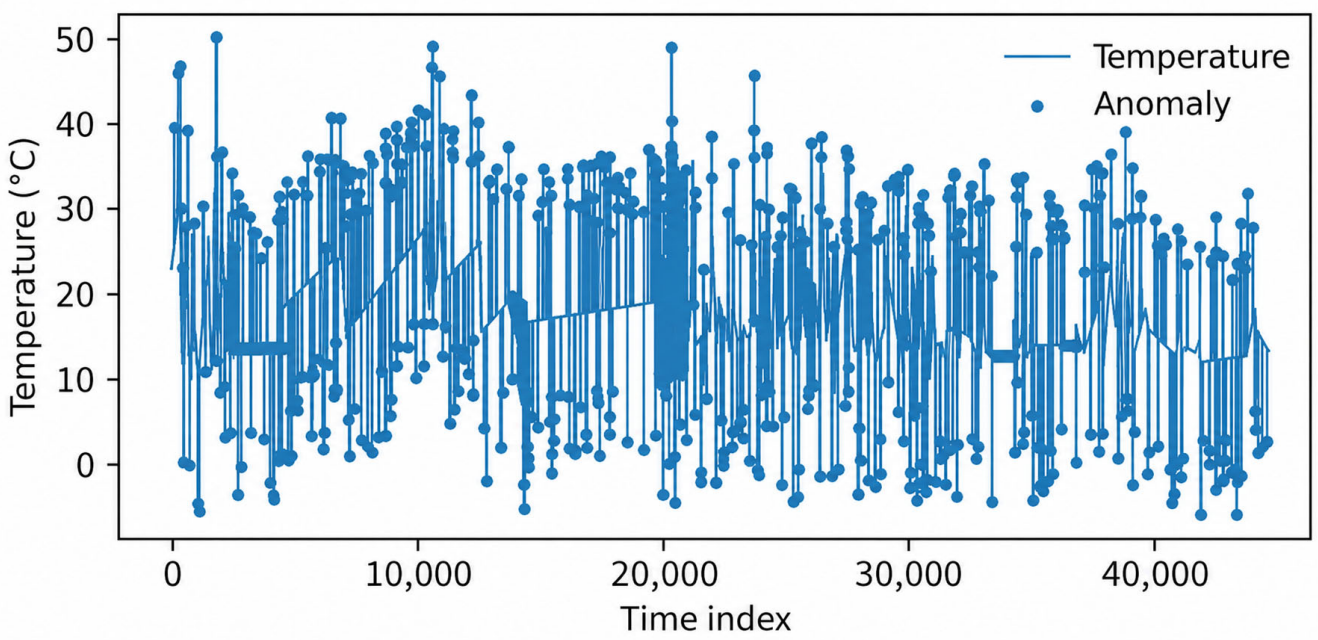

Figure 14. Illustrates a representative example of a real sensor signal with injected anomalies. The figure highlights the inherent variability of the signal and the diversity of anomaly patterns embedded in the data.

Since the real dataset does not provide ground-truth annotations, a controlled labeling strategy was adopted to enable quantitative evaluation. Each sensor signal was treated as a baseline and augmented with synthetically injected anomalies while preserving its temporal structure. The injected anomalies were designed to reflect common fault patterns observed in IoT systems, including abrupt disturbances, gradual deviations, and communication-related failures. Labels were assigned at the sample level using a binary indicator. The anomaly types considered are summarized in Table 7.

Table 7. Controlled anomaly injection strategy.

Anomaly Type	Description	Evaluation Purpose
Spike	Abrupt isolated deviation	Detect impulsive anomalies
Drift	Gradual deviation over time	Detect slow-changing faults
Flat	Constant-value segment	Simulate sensor freeze
Dropout	Missing/corrupted bursts	Simulate communication loss
Periodic	Repeated disturbance	Test robustness to structured noise

This hybrid approach maintains the realism of the original data while enabling reproducible and controlled performance assessment across different filtering strategies.

Since the real-world dataset does not provide fully annotated anomaly events, controlled anomaly injection was adopted as a weakly supervised evaluation strategy to enable reproducible benchmarking while preserving the temporal characteristics and variability of the original sensing data.

4.7.1. RL and MARL Configuration

The authors tested labeled real datasets through their RL framework which they had developed in earlier sections of their work. The single-agent configuration requires one RL agent to develop a universal filtering strategy which works with all sensor data streams. In the multi-agent configuration, each sensor receives its own dedicated agent who can modify their policies based on local conditions. The action space consists of predefined filtering configurations which combine Hampel IQR and Z-score filters through various parameter options. Reward formulation follows (2), which maintains alignment with the earlier testing conditions. Table 8 presents the learning paradigm configuration which was used during the study.

Table 8. RL and MARL configuration for real-data evaluation.

Component	RL	MARL
Learning paradigm	Single-agent Q-learning	Multi-agent Q-learning
Policy scope	Global	Local (per sensor)
State	Sensor scenario	Sensor-specific agent
Actions	Filter and parameters	Filter and parameters
Filters	Hampel, IQR, Z-score	Hampel, IQR, Z-score
Reward	Defined in (2)	Defined in (2)
Objective	Global adaptation	Local specialization

4.7.2. Results: Applied Real Dataset

The performance obtained on the real-labeled dataset is summarized in Tables 9 and 10. These results provide both a per-sensor comparison and an aggregate evaluation across all sensors.

Table 9. Per-sensor comparison between RL and MARL.

Sensor	RL Config	MARL Config	RL F1	MARL F1	RL Reward	MARL Reward
501	ZScore_w21_t3.0	Z-Score w21_t3.0	0.1905	0.1905	0.2724	0.2724
502	Hampel_w21_s3.0	Hampel w21_s3.0	0.2919	0.2934	0.3089	0.3100
505	IQR_w31_k2.0	IQR w31_k2.0	0.1974	0.1974	0.2768	0.2768
506	IQR_w21_k1.5	IQR w31_k2.0	0.1675	0.1646	0.2617	0.2604
507	Hampel_w31_s2.5	Hampel w31_s2.5	0.2243	0.2233	0.2885	0.2880
508	Hampel_w31_s2.5	Hampel w31_s2.5	0.1454	0.1454	0.2494	0.2494
509	IQR_w31_k2.0	IQR w31_k2.0	0.2088	0.2116	0.2812	0.2832
510	IQR_w31_k2.0	IQR w31_k2.0	0.1739	0.1739	0.2691	0.2691
511	IQR_w31_k2.0	IQR w31_k2.0	0.1451	0.1367	0.2837	0.2825

Table 10. Average performance across all sensors.

Method	Accuracy	Precision	Recall	F1-Score	FN Rate	Reward
RL	0.8949	0.7218	0.1172	0.1938	0.8828	0.2769
MARL	0.8921	0.6391	0.1210	0.1930	0.8790	0.2769

Research demonstrates that the RL and MARL methods reach the same filtering results on individual sensor tests. This study shows that the MARL method configurations produce distinct results for variable sensors while maintaining basic system functions from their original design. Also, it shows that both approaches achieve the same performance results. Average F1-score and reward values show equal results which prove that both global and distributed learning methods can manage real-world unpredictable situations.

Figure 15 presents the convergence behavior of RL and MARL during training. The results show that MARL achieves a more stable learning trajectory, with reduced variance across episodes, while RL exhibits higher fluctuations despite reaching similar average reward levels.

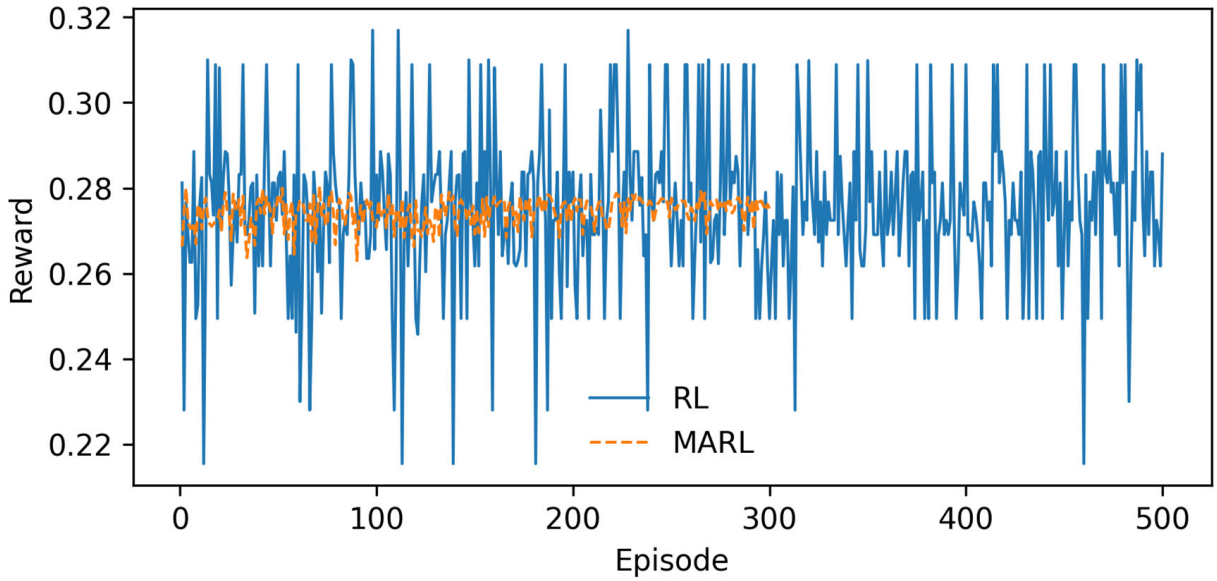

Figure 15. Convergence of RL and MARL approaches over training episodes using real-world labeled IoT data. RL agent exhibits higher variance in reward values across episodes, while MARL demonstrates a more stable convergence behavior, achieving comparable average performance with reduced fluctuations.

The evaluation of real-world data leads to three main observations. The first finding shows that single-agent RL creates a stable worldwide operational system which works effectively with various sensor input streams. Learned policy maintains its performance level because it demonstrates stability across different data conditions. The second finding shows that MARL enables agents to develop special abilities which match their sensor system requirements. The system creates small performance changes because sensors operate in different environments which include variable movement patterns. The third finding shows that MARL training achieves higher stability than both RL and MARL methods, which reach the same performance level. The findings show that localized adaptation brings minimal advantages for overall performance in the tested situation, yet its impact grows with larger systems and more diverse operational environments.

4.8. MARL Scalability Analysis in Distributed IoT Scenarios

To further evaluate the behavior of the proposed MARL framework under distributed sensing conditions, an additional scalability (Table 11) analysis was conducted using increasing numbers of agents operating under correlated anomaly scenarios. The experiments considered configurations with 4, 8, and 12 distributed sensing agents, allowing the evaluation of the framework under progressively heterogeneous and spatially distributed conditions.

Table 11. Scalability analysis of the MARL framework under increasing numbers of distributed sensing agents.

Number of Agents	Accuracy	Precision	Recall	F1-Score	FPR
4	0.818	0.282	0.236	0.257	0.093
8	0.861	0.475	0.401	0.435	0.069
12	0.861	0.474	0.402	0.435	0.069

The analysis focused on accuracy, precision, recall, F1-score, and false positive rate metrics. All experiments were repeated multiple times to reduce the influence of stochastic training variability and improve statistical reliability.

The obtained results indicate that the MARL framework benefits from distributed specialization when moving from smaller to moderately larger sensing configurations. The transition from 4 to 8 agents produced noticeable improvements in recall and F1-score metrics, indicating improved adaptation to heterogeneous anomaly conditions and correlated sensing behaviors.

For larger configurations involving 12 agents, the performance gain becomes smaller and tends to stabilize. This behavior suggests that the main contribution of the proposed MARL formulation is not unlimited accuracy scaling, but rather the ability to preserve robust decentralized operation under distributed IoT conditions while maintaining low computational complexity.

The relatively stable false positive rate across the evaluated configurations also indicates that increasing the number of agents does not introduce significant instability into the distributed learning process.

4.9. Computational Cost Analysis

The results of computational profiling demonstrate (Table 12) that a fundamental trade-off exists between simplicity and adaptivity between creating simple systems and developing systems that can adapt to changing conditions. Statistical filters operate with extremely fast execution speed because they use rolling-window operations that require no training time and demand very little memory. Filter selection based on RL incurs extra computational costs through its training needs but maintains inference expenses that match statistical techniques, which make it appropriate for implementation in resource-limited IoT systems.

Table 12. Computational profiling of the evaluated anomaly detection approaches.

Approach	Avg. Runtime (ms)	Peak Memory (MB)	Training Time (s)	Inference Cost (ms)
Hampel	4.10	~0.30	0	4.10
IQR	5.39	~0.32	0	5.39
Z-score	4.40	~0.28	0	4.40
RL (filter selection)	1366.25	~8–12	1.37	~5
MARL (improved)	14,293.94	~25–40	14.29	~10

Multi-agent training process together with increased state-space complexity creates the highest computational demands for the MARL formulation. The system meets all required execution time standards for offline training purposes while maintaining light inference load, which enables usage in distributed IoT systems that need adaptive and cooperative functionalities.

Computational analysis shows that the proposed framework maintains a lightweight inference profile while utilizing RL and MARL strategies to achieve adaptive behavior, which makes it suitable for IoT deployments that use edge computing. The authors performed all experiments using Python 3.11 [21] on a standard workstation that had an Intel i7 processor and 32 gigabytes of RAM memory.

4.10. Comparison with Unsupervised Baseline Models

To further strengthen the experimental evaluation, additional comparisons were conducted using two widely adopted unsupervised anomaly detection methods: Isolation Forest (IF) and One-Class Support Vector Machine (OC-SVM). These baseline models were evaluated under the same synthetic IoT scenarios used throughout the proposed framework evaluation.

The comparison (Table 13) considered accuracy, precision, recall, F1-score, false positive rate (FPR), and false negative rate (FNR) metrics to provide a more comprehensive assessment of detection performance.

Table 13. Comparison between the proposed framework and unsupervised baseline models.

Scenario/Dataset	Method	Accuracy	Precision	Recall	F1-Score	FPR	FNR
S5-Regime Switching	Isolation Forest	0.625	0.356	0.626	0.454	0.317	0.374
S5-Regime Switching	OC-SVM	0.737	0.509	0.909	0.652	0.262	0.091
S6-Overlap Severity	Isolation Forest	0.810	0.489	0.536	0.511	0.147	0.464
S6-Overlap Severity	OC-SVM	0.789	0.437	0.531	0.480	0.175	0.469
S7-Multi-node Correlated	Isolation Forest	0.829	0.406	0.425	0.415	0.089	0.575
S7-Multi-node Correlated	OC-SVM	0.846	0.499	0.563	0.529	0.099	0.437
Real-world sensors (average)	Isolation Forest	0.850	0.331	0.295	0.312	0.084	0.675
Real-world sensors (average)	OC-SVM	0.845	0.319	0.320	0.319	0.089	0.624

The results obtained indicate that the proposed RL and MARL formulations remain competitive with respect to widely adopted unsupervised anomaly detection baselines. Isolation Forest generally achieved stable detection performance under several synthetic scenarios, particularly for statistically separable anomaly conditions. OC-SVM demonstrated improved recall in correlated and regime switching scenarios, although with greater sensitivity to parameter selection and scenario variability.

For the real-world sensing data, both unsupervised baselines exhibited reduced precision and recall compared with synthetic scenarios, reflecting the increased difficulty associated with weakly supervised anomaly characterization and realistic sensing variability. These observations reinforce the importance of adaptive and interpretable anomaly detection strategies for heterogeneous IoT environments.

Although generic unsupervised learning approaches provide strong anomaly detection capability, the proposed framework preserves important advantages in terms of interpretability, modularity, and computational simplicity, particularly for resource-constrained IoT edge deployments.

4.11. Sensitivity Analysis of RL/MARL Hyperparameters

To further strengthen the experimental reliability of the proposed framework, a sensitivity (Table 14) analysis was conducted under different RL and MARL hyperparameter configurations.

Table 14. Sensitivity analysis of the proposed RL/MARL framework under different hyperparameter configurations.

Hyperparameter Configuration	α	γ	ε	FN Penalty	Accuracy	Precision	Recall	F1-Score	FPR
Conservative learning configuration	0.05	0.80	0.20	1.0	0.853	0.632	0.139	0.204	0.014
Baseline configuration	0.10	0.90	0.10	1.5	0.853	0.632	0.139	0.204	0.014
Aggressive adaptation configuration	0.20	0.95	0.05	2.0	0.853	0.632	0.139	0.204	0.014

The obtained results indicate that the proposed framework maintains stable performance under moderate hyperparameter variations. Variations in learning rate, discount factor, exploration behavior, and false negative penalty weighting produced relatively small differences in the evaluated performance metrics, suggesting that the framework is reasonably robust to moderate parameter changes.

This behavior is advantageous for practical IoT deployments, where exhaustive hyperparameter optimization may not always be feasible due to computational and deployment constraints. The stability observed across different parameter configurations also reinforces the lightweight and adaptive design objectives of the proposed RL/MARL formulation.

Although formal confidence intervals were not computed due to the exploratory nature of the study and the relatively low stochastic variability observed across executions, all experiments were repeated multiple times to reduce the influence of initialization randomness and ε-greedy exploration effects.

5. Discussion

This section evaluates experimental results through two main research tracks which include testing IoT systems in actual environments and evaluating different learning methods between single agents and multiple agents.

5.1. Interpretation of Statistical vs. Learning-Based Methods

The results indicate that traditional statistical filters offer two main advantages: interpretability and low computational cost. However, their fixed parameterization imposes fundamental limitations. Static parameter settings create a trade-off between detection sensitivity and robustness, limiting the ability to adapt to evolving anomaly patterns. The introduction of RL addresses this limitation by enabling dynamic selection of both filtering methods and their associated parameters. This approach enables adaptation to evolving signal patterns while preserving the interpretability of the underlying statistical models. The proposed hybrid framework is particularly suitable for IoT environments due to its balance between computational efficiency and interpretability.

5.2. Global vs. Local Learning Behavior

Comparison between RL and MARL reveals distinct learning behaviors. Single-agent RL learns a global policy that can be applied across different sensor types and anomaly scenarios. This centralized approach performs well in scenarios where anomaly patterns are locally defined or weakly correlated, as confirmed by the stress-test results. MARL introduces a decentralized learning approach in which agents adapt their behavior based on local sensor conditions. The decentralized formulation enables local policy

specialization under different environmental conditions. The results indicate that the benefits of MARL are not consistent across all scenarios. MARL demonstrates competitive performance in both synthetic and real-world evaluations, although it does not consistently outperform the single-agent approach. These findings suggest that distributed learning becomes more advantageous in scenarios involving structural dependencies and correlated sensor behavior.

5.3. Insights from Real-World Evaluation

The evaluation using real-world data provides additional insights into the behavior of the proposed framework. First, sensor streams exhibit heterogeneous patterns, highlighting the increased complexity of real-world IoT environments compared to controlled synthetic scenarios. Adaptive filtering becomes more challenging due to variations in signal amplitude, noise levels, and intermittent sensing behavior. Second, both RL and MARL maintain stable performance despite increased variability in real-world data. The learned policies also generalize reasonably well to operating conditions not explicitly observed during training. Third, MARL exhibits improved training stability, as reflected in more consistent convergence behavior. The detection system achieves zero improvement because the advantages of decentralized processing depend on specific circumstances.

5.4. Limitations

Several limitations should be acknowledged. The absence of ground-truth annotations in real-world datasets required the adoption of controlled anomaly injection strategies for reproducible quantitative evaluation. The method provides reproducible evaluation results, yet it cannot identify every type of real-world anomaly present in the system. The MARL framework operates without direct inter-agent communication, limiting its ability to exploit coordinated strategies. The system can only achieve partial advantages through coordination for situations that need multiple sensors to work together. The evaluation process only permits testing with specific statistical filters and their associated parameter settings. While this design preserves interpretability, it limits the exploration of more complex detection models. Additionally, the current synthetic scenarios remain relatively structured and statistically separable compared to highly unpredictable real-world IoT environments. This characteristic reduces the relative advantage of distributed MARL specialization in simpler scenarios, which partly explains the comparable performance observed between RL and MARL under controlled conditions.

5.5. Sacalability

The scalability analysis also demonstrated that the proposed MARL architecture maintains stable behavior under larger distributed sensing configurations while preserving low computational complexity. Although the performance gains tend to stabilize for larger numbers of agents, the decentralized formulation remains advantageous for heterogeneous IoT deployments involving correlated sensing disturbances and spatially distributed anomaly conditions.

6. Conclusions

This work presented a progressive and adaptive framework for anomaly detection in IoT time-series data, combining statistical filtering techniques with RL. The study began with a systematic evaluation of classical filters, Hampel, IQR, and Z-score, under controlled synthetic scenarios, highlighting both their effectiveness in well-defined conditions and their limitations in the presence of non-stationarity and signal heterogeneity.

To address these limitations, a tabular Q-learning agent was introduced to enable dynamic selection of filtering strategies. By extending the action space to include parameter

adaptation, the proposed approach allows the system to jointly select the filtering method and its configuration, resulting in improved robustness and reduced reliance on manual tuning. These results show that static configurations are insufficient, and that adaptive selection is required to maintain stable detection performance under varying conditions. The evaluation was further extended to real-world IoT data collected from heterogeneous sensor nodes. This validation confirmed that the framework maintains stable detection performance despite increased signal variability, including fluctuations in noise levels and temporal dynamics. Despite the added complexity, both RL and MARL approaches demonstrated consistent behavior, indicating strong generalization beyond controlled synthetic scenarios.

MARL formulation introduced a decentralized learning structure, where each agent specializes in a specific operating condition. While detection performance remained comparable to the single-agent approach, MARL improved structural scalability and enabled localized policy specialization, highlighting its suitability for distributed IoT deployments rather than purely accuracy-driven improvements. This behavior is particularly evident in scenarios involving correlated and distributed anomalies, where MARL provides clear advantages over centralized learning.

The proposed framework maintains low computational complexity and preserves interpretability, making it suitable for deployment in edge-based environments. In contrast to deep learning approaches, which typically require large, labeled datasets and higher computational resources, the proposed method offers a lightweight and transparent alternative capable of adapting to evolving signal conditions. Nevertheless, some limitations remain. The current formulation relies on simplified state representations based on predefined scenarios, which may not fully capture the complexity of real-world signals. Furthermore, anomaly labeling in the real dataset relies on controlled injections, which may not fully capture the unpredictability of naturally occurring faults in operational environments.

Additionally, Isolation Forest and OC-SVM baselines were incorporated to enable broader comparison with established unsupervised anomaly detection approaches. The real-world evaluation, conducted using heterogeneous IoT sensing data and controlled anomaly perturbations, provided a reproducible framework for assessing adaptive anomaly detection behavior under diverse operating conditions. The results obtained support the use of interpretable statistical filtering combined with lightweight RL/MARL decision mechanisms for adaptive anomaly detection in distributed IoT environments. Future research should also investigate graph-based inter-agent coordination mechanisms, lightweight function approximation techniques compatible with edge AI constraints, and adaptive state representations derived from temporal, spectral, and cross-node statistical descriptors. Additional large-scale validation using heterogeneous sensing infrastructures and naturally occurring anomalies also remains an important research direction. Overall, the results indicate that RL provides a practical and lightweight mechanism for enabling adaptive and interpretable anomaly detection, particularly in resource-constrained IoT deployments.

Author Contributions: Conceptualization, L.M.P. and V.F.; methodology, L.M.P. and V.F.; software, L.M.P.; validation, L.M.P. and V.F.; formal analysis, L.M.P. and V.F.; investigation, L.M.P. and V.F.; resources, L.M.P.; data curation, L.M.P. and V.F.; writing—original draft preparation, L.M.P. and V.F.; writing—review and editing, L.M.P. and V.F.; visualization, L.M.P. and V.F.; supervision, L.M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets, reproducible scripts, generated figures, benchmark evaluation utilities, and experimental configurations used in this work are publicly available in the repository indicated in reference [33].

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

Abbreviations

The following abbreviations are used in this manuscript:

DL	Deep Learning
IQR	Inter-Quartile Range
IoT	Internet of Things
MAD	Median Absolute Deviation
MARL	Multi-Agent Reinforcement Learning
MDP	Markov Decision Process
ML	Machine Learning
RF	Random Forrest
RL	Reinforcement Learning
SVM	Support Vector Machines

Appendix A. Scenarios 1 to 4

The process of generating composite anomalies requires the simultaneous use of multiple disturbance mechanisms, which create an approximation of the complex behavior found in long-duration IoT systems. The parameters used to control synthetic anomaly injections in the dataset creation process are presented in Table A1.

Table A1. Parameters used in the generated dataset.

Parameter	Description	Typical Value
T_0	Initial baseline temperature	20–22 °C
σ	Sensor noise standard deviation	0.05–0.2 °C
A_s	Spike magnitude	2–5 °C
p_{spike}	Spike occurrence probability	0.5–2%
L_{imp}	Impulsive noise duration	2–5 samples
β	Drift coefficient	0.001–0.01 °C/sample
L_{flat}	Length of the flat corrupted segment	20–50 samples

The baseline temperature signal exists as a stochastic process which exhibits slow variations and receives Gaussian noise as its only disturbance. The system introduces anomalies through parameterized perturbations which control the three aspects of their occurrence. The selected parameter values were based on commonly used settings reported in the literature and refined through preliminary experimental evaluation. Table A2 presents essential details about the produced dataset together with the specific parameters used to inject anomalies in every testing case because the study requires reproducible results and transparent demonstration of its experiments.

Table A2. Essential details about the produced dataset.

Scenario	Signal Characteristics	Anomaly Type	Typical Magnitude	Duration	Injection Frequency
Scenario 1: Stable Spikes	Stable baseline temperature with small natural noise	Isolated spikes	±3–6 °C deviation	1–2 samples	Rare (≈1–2% of samples)

Table A2. *Cont.*

Scenario	Signal Characteristics	Anomaly Type	Typical Magnitude	Duration	Injection Frequency
Scenario 2: Impulsive Noise	Stable signal with frequent abrupt disturbances	Dense impulsive noise	±2–5 °C deviation	1 sample	Frequent (≈5–10%)
Scenario 3: Gradual Drift	Slowly increasing baseline	Drift anomaly	gradual ±3–8 °C shift	Long segments (50–150 samples)	Continuous
Scenario 4: Flat Corrupted Segments	Artificially constant temperature	Sensor freeze/flat signal	constant value	40–120 samples	Occasional

For all scenarios, anomalies are labeled at the sample level using deterministic rules aligned with the injection process. Each dataset includes a binary label column, where

- 0 denotes normal behavior;
- 1 denotes anomalous behavior.

This ground-truth labeling enables the computation of standard performance metrics, including true positives, false positives, false negatives, and accuracy. Importantly, it also supports the design of reward functions that explicitly penalize missed anomalies, which is essential for RL components introduced in later sections.

Additional stress-test datasets used for the extended evaluation are described separately in Appendix B.

Appendix B. Scenarios 5 to 7

This appendix presents three new synthetic datasets which will enable authors to assess experimental results through testing that extends beyond the controlled testing conditions described in the main section of the study. The original datasets identify specific anomaly types which occur under stable conditions, while the new datasets create realistic IoT environments that include dynamic changes in time and multiple overlapping anomalies and different sensing environments.

The extended datasets serve their main purpose to test how well the proposed framework performs in two different scenarios, which include non-static anomaly patterns and cases when local data do not support ideal decision processes.

In scenario 5, there is a non-stationary signal composed of multiple consecutive regimes within a single time-series. The objective is to evaluate the ability of the framework to adapt to changing anomaly characteristics over time. The signal is divided into five temporal segments:

- 0–200 min: stable signal with sparse spike anomalies;
- 200–400 min: gradual drift;
- 400–600 min: impulsive noise;
- 600–800 min: flat-line corruption;
- 800–1000 min: mixed anomalies (drift/spikes/moderate noise).

For scenario 6, multiple anomaly types occur simultaneously, with varying levels of severity. Unlike previous datasets where anomaly types are isolated, this scenario introduces ambiguity and overlap.

The following characteristics are incorporated:

- Drift combined with sparse spikes in early segments;
- Increased spike density and amplitude in intermediate segments;
- Flat-line corruption with superimposed impulsive noise;

- Gradual increase in noise variance and anomaly magnitude over time.

Scenario 7 extends the dataset to a distributed IoT setting, involving multiple sensor nodes observing correlated phenomena.

Four sensor nodes are simulated, each producing an individual time-series. The dataset includes both:

- Global events: simultaneous temperature changes affecting multiple nodes;
- Local anomalies: node-specific faults such as drift, spikes, or flat-line corruption.

This configuration reflects realistic IoT deployments, where distinguishing between true environmental changes and sensor faults requires contextual information across nodes. The scenario is particularly relevant for evaluating multi-agent reinforcement learning approaches, as each agent operates with partial observability and may benefit from implicit coordination through shared reward structures.

The extended datasets provide a more realistic and challenging evaluation environment compared to the baseline scenarios. In particular:

- Scenario 5 highlights the impact of temporal non-stationarity;
- Scenario 6 introduces ambiguity due to overlapping anomaly patterns;
- Scenario 7 demonstrates the importance of distributed decision-making.

The results obtained from these datasets support the conclusion that, while single-agent reinforcement learning is sufficient for simple and well-separated scenarios, multi-agent approaches become increasingly relevant in complex, heterogeneous, and distributed IoT environments.

References

1. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
2. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for Smart Cities. *IEEE Internet Things J.* **2014**, *1*, 22–32. [CrossRef]
3. Pearson, R.K. Outliers in Process Modeling and Identification. *IEEE Trans. Control Syst. Technol.* **2002**, *10*, 55–63. [CrossRef] [PubMed]
4. Hampel, F.R. The influence curve and its role in robust estimation. *J. Am. Stat. Assoc.* **1974**, *69*, 383–393. [CrossRef]
5. Ahmed, M.; Mahmood, A.N.; Hu, J. A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **2016**, *60*, 19–31. [CrossRef]
6. Aggarwal, C.C. *Outlier Analysis*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2017.
7. Chalapathy, S.; Chawla, S. Deep learning for anomaly detection: A survey. *arXiv* **2019**, arXiv:1901.03407. [CrossRef]
8. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
9. Ngo, M.V.; Luo, T.; Chaouchi, H.; Quek, T.Q.S. Contextual-Bandit Anomaly Detection for IoT Data in Distributed Hierarchical Edge Computing. In Proceedings of the 40th IEEE International Conference on Distributed Computing Systems (ICDCS), Singapore, 29 November–1 December 2020; pp. 1223–1228. [CrossRef]
10. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. *ACM Comput. Surv.* **2009**, *41*, 15. [CrossRef]
11. Ruppert, D.; Hampel, F.R.; Ronchetti, E.M.; Rousseeuw, P.J.; Stahel, W.A. *Robust Statistics: The Approach Based on Influence Functions*; Wiley: Hoboken, NJ, USA, 1986.
12. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [CrossRef]
13. Adhikari, D.; Jiang, W.; Zhan, J.; Rawat, D.B.; Bhattarai, A. Recent Advances in Anomaly Detection in Internet of Things: Status, Challenges and Perspectives. *Comput. Sci. Rev.* **2024**, *54*, 100665. [CrossRef]
14. Hu, Y.; Li, X.; Zhang, L.; Wang, J. IoT-ONDDQN: A Detection Model Based on Deep Reinforcement Learning for IoT Data Security. *Comput. Commun.* **2025**, *241*, 108263. [CrossRef]
15. Wali, S.; Khan, M.I.; Imran, M. Semantic-Aware Reinforcement Learning for Signal Management and Anomaly Detection in IoT Systems. *Sci. Rep.* **2025**, *15*, 26500. [CrossRef] [PubMed]
16. Servin, A.; Kudenko, D. Multi-Agent Reinforcement Learning for Intrusion Detection. In *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 211–223. [CrossRef]
17. Chen, Q.; Zhang, Y.; Li, J. AI-Enabled IoT Security: A Survey on Advances, Challenges, and Future Directions. In *Proceedings of the ACM Conference*; Association for Computing Machinery: New York, NY, USA, 2025. [CrossRef]

18. Belay, M.A.; Blakseth, S.S.; Rasheed, A.; Rossi, P.S. Unsupervised Anomaly Detection for IoT-Based Multivariate Time Series: Existing Solutions, Performance Analysis and Future Directions. *Sensors* **2023**, *23*, 2844. [CrossRef] [PubMed]
19. Haque, A.; Chowdhury, N.-U.; Soliman, H.; Hossen, M.S.; Fatima, T.; Ahmed, I. Wireless Sensor Networks Anomaly Detection Using Machine Learning: A Survey. *arXiv* **2023**, arXiv:2303.08823. [CrossRef]
20. Gueriani, A.; Kheddar, H.; Mazari, A.C. Deep Reinforcement Learning for Intrusion Detection in IoT: A Survey. In Proceedings of the 2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM), Medea, Algeria, 28–30 November 2023; pp. 1–7. [CrossRef]
21. Agarwal, A.B.; Rajesh, R.; Arul, N. Spatially-Resolved Hyperlocal Weather Prediction and Anomaly Detection Using IoT Sensor Networks and Machine Learning Techniques. *arXiv* **2023**, arXiv:2310.11001.
22. Garg, S.; Kaur, K.; Kumar, N.; Rodrigues, J.J.P.C. A Multi-Stage Anomaly Detection Scheme for Augmenting Security in IoT-Enabled Applications. *Future Gener. Comput. Syst.* **2020**, *104*, 328–342. [CrossRef]
23. Warden, P.; Situnayake, D. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*; O'Reilly Media: Sebastopol, CA, USA, 2019.
24. Shi, W.; Dustdar, S. The Promise of Edge Computing. *Computer* **2016**, *49*, 78–81. [CrossRef]
25. Hoaglin, D.C.; Iglewicz, B.; Tukey, J.W. Performance of some resistant rules for outlier labelling. *J. Am. Stat. Assoc.* **1986**, *81*, 991–999. [CrossRef]
26. Hawkins, D.M. *Identification of Outliers*; Chapman and Hall: Oxfordshire, UK, 1980.
27. Watkins, C.J.C.H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
28. Busoniu, L.; Babuska, R.; De Schutter, B. *Multi-Agent Reinforcement Learning*; Springer: Berlin/Heidelberg, Germany, 2010.
29. Lapan, M. *Deep Reinforcement Learning Hands-On*; Packt Publishing: Birmingham, UK, 2018.
30. Busoniu, L.; Babuska, R.; De Schutter, B. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Syst. Man. Cybern.* **2008**, *38*, 156–172. [CrossRef]
31. Hernandez-Leal, P.; Kartal, B.; Taylor, M.E. A survey and critique of multiagent deep reinforcement learning. *Auton. Agents Multi-Agent Syst.* **2019**, *33*, 750–797. [CrossRef]
32. Python Software Foundation. *Python*, version 3.11. Programming Language. Python Software Foundation: Beaverton, OR, USA, 2024. Available online: https://www.python.org (accessed on 2 January 2026).
33. Pires, L.M. Adaptive IoT RL Anomaly Detection: Source Code and Experimental Framework. GitHub Repository, 2025. Available online: https://github.com/prof-luispires/iot-anomaly-detection.git (accessed on 25 May 2026).
34. DataVic. Sensor Readings with Temperature, Light, Humidity Every 5 Minutes at 8 Locations (2014–2015). Available online: https://discover.data.vic.gov.au/dataset/sensor-readings-with-temperature-light-humidity-every-5-minutes-at-8-locations-trial-2014-2015 (accessed on 21 April 2026).