ISSN 2007-9737

Optimizing Patient Monitoring with Reinforcement Learning Based Context-Aware Healthcare Formalism

Hafiz Mahfooz Ul Haque[1], Abdullah[2,3], Faiza Tariq[4,5], Shahid Yousaf[6], José Luis Oropeza-Rodríguez[2,*]

[1] University of Central Punjab,
Department of Software Engineering, Faculty of IT & CS,
Pakistan

[2] Instituto Politecnico Nacional,
Centro de Investigacion en Computacion,
Mexico

[3] Bahria University,
Department of Computer Science,
Pakistan

[4] University of Glasgow,
James Watt School of Engineering,
UK

[5] University of Education,
Department of Information Sciences,
Pakistan

[6] University of Lahore,
Department of Computer Science & IT,
Pakistan

mahfooz.haque@ucp.edu.pk, abdullah2025@cic.ipn.mx,
faiza.tariq@glasgow.ac.uk, shahid.yousaf@uol.edu.pk, joropeza@cic.ipn.mx

Abstract. The rapid evolution of smart devices and software technologies has transformed modern computing, enabling seamless mobility, pervasive service accessibility, and context-sensitive interactions. With the advent of the Internet of Things (IoT), healthcare systems have gained unprecedented capabilities in monitoring, reliability, and security. However, managing distributed systems in highly decentralized settings while ensuring intelligent, adaptive, and safe decision-making remains a critical challenge. To address this gap, we propose a context-aware healthcare formalism based on hybrid reinforcement learning for real-time patient monitoring. The framework is modeled as a Markov Decision Process (MDP), where smart devices equipped with embedded sensors acquire and analyze contextual information, communicate with other agents, and adapt dynamically to achieve healthcare goals. The proposed Smart Healthcare Context-Aware System (SHCS) integrates reinforcement learning with rule-based reasoning to balance safety guarantees and operational efficiency. A case study and prototype implementation demonstrate its feasibility, and the experimental results show that the hybrid system achieves a true positive rate of 99.99%, reducing the false positive rate by 88% compared to the baselines with only rules of 1. 85% <0.01. 22%. It also reduces caregiver workload with a reduction in alerts per episode of 71% and improves response time by

14% (2.1 s 1.8 s). These findings are further supported through formal verification in PRISM, ensuring that safety restrictions are never violated. Collectively, the results underscore the potential of hybrid reinforcement learning for building trustworthy, scalable, and intelligent healthcare monitoring systems deployable in real-world clinical settings.

Keywords. Hybrid reinforcement learning, context-aware healthcare, multi-agent systems, Markov decision process (MDP), rule-based decision support.

1 Introduction

The last decade has witnessed a rapid transformation in healthcare systems driven by the convergence of artificial intelligence (AI), Internet of Things (IoT), and ubiquitous computing [8, 17]. The growing prevalence of wearable sensors, mobile devices, and smart medical infrastructures has allowed continuous monitoring of patient vital signs, providing opportunities for early diagnosis, personalized care, and reduced clinical burden [34].

However, these advancements have also introduced challenges related to scalability, real-time decision making, data reliability, and patient safety in highly dynamic and decentralized environments [14]. In particular, ensuring that healthcare monitoring systems can adapt to changing patient conditions while guaranteeing timely and safe interventions remains a pressing research problem.

Reinforcement learning (RL) has emerged as a promising paradigm for sequential decision-making under uncertainty, enabling systems to learn adaptive policies through interactions with their environment. In healthcare, RL has been applied to tasks such as treatment optimization, glucose regulation, and personalized intervention planning. Yet, pure RL approaches face inherent limitations in safety-critical domains: exploratory actions may lead to dangerous delays, missed detections, or resource misallocations [18, 35, 42].

Conversely, rule-based systems provide strict guarantees of safety but are often conservative, leading to excessive false alarms and caregiver fatigue. Bridging the gap between safety assurance and adaptive efficiency, therefore calls for hybrid solutions that integrate the strengths of both paradigms [9, 5].

Context-awareness plays a pivotal role in enabling intelligent healthcare monitoring. By capturing information about the patient's physical state, environmental conditions, and temporal factors, context-aware systems can tailor responses to evolving situations and support more precise interventions. When combined with RL and multi-agent coordination, context-awareness allows healthcare systems not only to react to abnormalities but also to anticipate risks and optimize care pathways in real time [35, 28].

This paper introduces a hybrid reinforcement learning based context-aware healthcare formalism designed to address two critical challenges in patient monitoring: safety and operational efficiency. The proposed framework leverages a Markov Decision Process (MDP) to model sequential healthcare decision-making, while employing multi-agent reinforcement learning (MARL) to enable distributed sensing, reasoning, and collaboration across smart devices [18].

A hybrid architecture ensures that deterministic safety rules enforce critical interventions, while RL agents optimize secondary objectives such as minimizing false alarms, reducing caregiver workload, and improving response times.

The main contributions of this work are as follows:

— A novel hybrid formalism that integrates reinforcement learning with rule-based reasoning to guarantee patient safety while enhancing operational efficiency.

— A multi-agent context-aware design for IoT-enabled healthcare, enabling distributed monitoring, knowledge sharing, and adaptive decision-making.

— Rigorous evaluation through simulation, formal verification in PRISM, and prototype testing, confirming near-perfect sensitivity and significant reductions in false alarms and resource costs.

The rest of the paper is structured as follows. Section 2 reviews background concepts and related work. Section 3 presents Multi-agent Reinforcement Learning based Smart Healthcare Context-Aware formalism. In Section 4, we illustrate the use of the proposed formalism using experimental setup and simulation. Section 5 reports results and formal verification; finally Section 6 concludes the paper with future research directions.

2 Preliminaries and Related Work

This section initially presents very briefly discussions on the core notions of context-awareness, context-aware systems and reinforcement learning paradigm. Then, we discuss the smart healthcare related work.

2.1 Context-Awareness and Context-aware System

Humans naturally possess context-awareness, adapting ideas and messages based on environmental situations, which directly influence intelligent behavior. The Literature categorizes context into user context as profile, location, and social situation, physical context as lighting, noise, temperature; and time context as time of day, week, or season. Context can also be classified into three key types: (a) object location, (b) surrounding objects, and (c) nearby resources, highlighting the importance of context-aware computing [39].

Context-aware computing enables software systems to adapt based on location, nearby entities, and environmental changes by leveraging context acquisition, context representation, context storage, and context interpretation [23, 33].

Context acquisition involves sensing raw data using sensors embedded in portable devices and surrounding environments, while context representation structures this data for efficient retrieval and sharing [38, 13].

The acquired context is then stored, not only preserving current data but also maintaining historical context to identify user preferences. To make this data meaningful, context interpretation converts raw information into high-level insights through context reasoning, which answers key questions such as "what," "who," "where," and "when."

This reasoning provides identification, activity recognition, and spatial-temporal insights, which ultimately guide context adaptation [27, 16]. Context adaptation tailors system responses to user needs using decision-making techniques, facilitating service discovery, service delivery, and service adaptation [29], thereby enhancing user experiences in dynamic environments.

2.2 Reinforcement Learning

Machine Learning (ML) has tackled diverse challenges over the past decade, including in healthcare, with the integration of Reinforcement Learning (RL) [44, 1].

RL, a powerful approach for sequential decision-making, operates in an interactive environment where agents learn optimal actions through rewards and penalties. Using models like Markov Decision Processes, agents refine their strategies based on experience [6]. In single-agent RL, each agent independently interacts with the environment to maximize rewards.

However, for complex and dynamic scenarios, Multi-Agent Reinforcement Learning (MARL) is more effective, enabling coordinated decision-making toward a shared goal. MARL facilitates policy learning by optimizing execution plans and exchanging contextual information among agents [40, 39].

MARL operates under three architectures: (a) decentralized, where agents learn and act independently, adjusting their policies based on experience; (b) centralized, where a central controller manages decision-making for optimal execution; and (c) centralized training with decentralized execution, where agents execute independently while sharing contextual data under a centralized framework [41, 20].

2.3 Related Work

Recent research has focused on developing smart healthcare systems by integrating IoT, context-awareness, and reinforcement learning. Several studies have proposed various methodologies for context-aware health monitoring.

In [22, 30], a wearable IoT-based system tracked vital signs using sensors and a Raspberry Pi, with data stored in the cloud. Similarly, [32, 4] introduced a hospital-based system using biometric sensors and Bluetooth, GSM, or Wi-Fi to transmit abnormal readings.

Raspberry Pi was also employed for real-time tracking and web alerts [31], while thermistors and infrared sensors were used for monitoring via Ethernet or USB [25]. A wireless ECG monitoring system with encryption for secure access was proposed in [24], and [15, 10] integrated MySQL and GSM modules for real-time ECG tracking and caregiver alerts.

A telemedicine system using ZigBee and Bluetooth was designed for wireless transmission and remote analysis through MATLAB [21]. Reinforcement learning has also been explored for blood glucose monitoring and physician policy optimization [43, 17]. Unlike these approaches, the proposed system enables context-aware agents to autonomously analyze patient conditions, make intelligent decisions, and act with minimal human intervention.

Recent advancements, such as a DRL-based mobile-fog-cloud framework with blockchain, optimize healthcare workflows through Markov Decision Processes, enhancing resource allocation and data integrity [36, 3]. SmartHealth uses machine learning for IoMT security, achieving 92% attack detection accuracy.

DRLBTS [26, 2] applies Q-learning and blockchain for dynamic healthcare workflow scheduling, while BDRL [7] improves the security and privacy of IIoT tasks.

3 Multi-agent Reinforcement Learning Formalism for Smart Healthcare System

This section presents a reinforcement learning-based, context-aware, multi-agent formalism for health monitoring, focusing on elderly or bedridden patients. The formalism is structured into two layers: (a) Agents' modeling and reasoning, where context-aware agents autonomously collect, process, and act on patient data, and (b) formal modelling and verification, which will be discussed later.

The modeling layer consists of n agents, $A_g = \{1, 2, \ldots, n\}$, where each agent $a_i \in A_g$ uses reinforcement learning (RL) to sense, learn, and adapt behavior for decision-making. Within this framework, agents collaborate to solve complex tasks, anticipate user needs, and operate effectively in dynamic environments.

RL-based reasoning mimics human decision-making by acquiring contextual information through sensors and inferring optimal actions to provide collaborative support. Each agent acts autonomously to solve specific tasks, while multiple agents coordinate to achieve more complex goals. Context-aware agents, as described by Chen et al.[12], are designed to anticipate user needs and act accordingly, making them highly effective in dynamic and unpredictable environments.

To adapt to changing contexts, agents revise their knowledge bases through a reward-penalty policy, selecting optimal execution plans, and updating their knowledge after each action. The agents' decision-making and sequential task execution are modeled using a Markov Decision Process (MDP), represented by the tuple (S, A, P, R, γ). Here, S denotes the set of states $S = \{s_1, s_2, \ldots, s_n\}$; A denotes the set of actions $A = \{a_1, a_2, \ldots, a_n\}$; P is the transition function, $P : S \times A$, mapping a current state (s_t) and action (a_t) to a next state (s_{t+1}); R is the reward function assigning values to transitions; and γ is the discount factor, ranging from 0 to 1, reflecting policy priorities and agent behavior.

A centralized architecture underpins the proposed formalism, where smart devices embedded with sensors acquire, process, and exchange

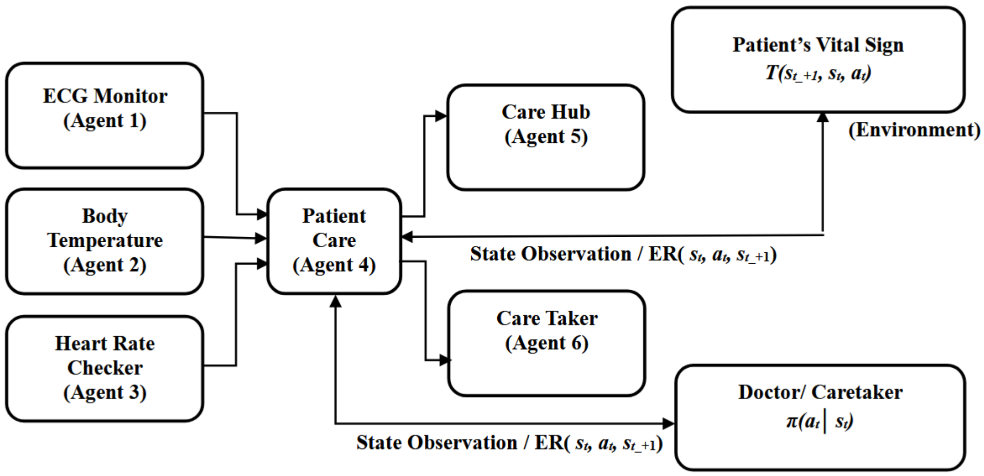

Fig. 1. RL based agent's interactive environment

contextual information to adapt their behavior and achieve target objectives. Since the system is deterministic and follows an optimistic reasoning approach, agents' action plans are formulated through state transitions governed by rewards and penalties. To support sequential decision-making in dynamic environments, the *SHCS* case study is also formulated as an MDP.

In this setup, an agent performs an action $a_t \in A$ in state $s_t \in S$ at time t, transitions to the next state $s_{t+1} \in S$, and receives a reward $r_{t+1}(s_{t+1}) \in R$. Agents periodically acquire sensor data and refine their actions based on accumulated rewards and penalties, gradually learning the optimal policy π^* that maximizes rewards across states and actions $\pi^*(S \times A)$. Figure 1 illustrates the reinforcement learning-based interaction of agents within the environment.

In this formalism, there are n agents, $A_g = \{1, 2, \ldots, n\}$ where each agent $a_i \in A_g$ is deployed in the system that uses the reinforcement learning (RL) based reasoning approach to achieve the desired goals. In the RL reasoning paradigm, artificial agents' working strategy is based on folk psychological notions of the human decision-making process.

The core objective of RL RL-based reasoning paradigm is to acquire contextual information from the environment using sensors and/or other systems and then perform reasoning to infer the desired goals in a customized setting for a collaborative decision support system. In the system, agents' activities are triggered by the set of actions on the state space that agents intend to perform.

So, acquiring/ perceiving the current facts/situation is added to the agent's knowledge base. These context-aware agents can be helpful to determine any context changes and the overall representation of context behavior.

The agents' knowledge base revision process is based on policies defined in terms of rewards and penalties. Based on the existing agents' policy, the system selects the optimal execution plans, and then agents are triggered to perform optimal actions to achieve the desired goals. After the execution of the agent's specific actions, agents update their knowledge base, and the next activation plan is executed according to the optimal execution plan.

So, agents' executions are traversed in the form of states and actions for complex problem-solving

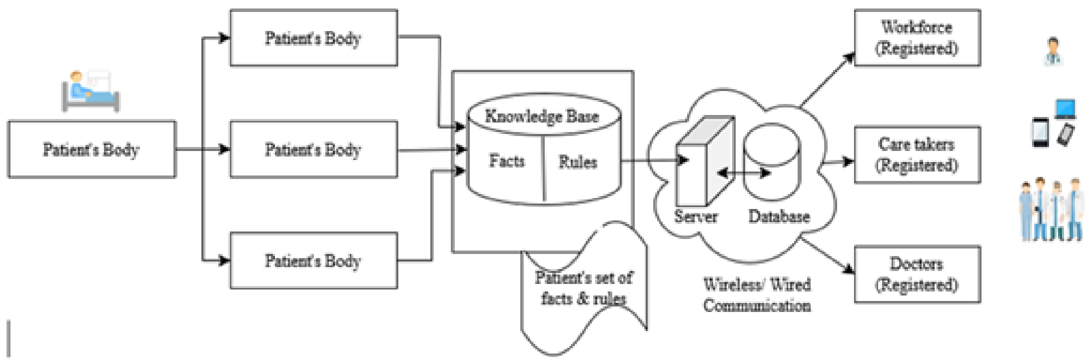

Fig. 2. Architecture of *SHCS* formalism

and sequential decision making, which is denoted by a mathematical structure known as the Markov Decision Process (MDP). In MDPs, decision-making strategies are developed using a specified set of rules known as policy. The policy provides guidelines on a set of actions that the agents perform to attain the desired objectives.

As the proposed system is deterministic and follows an optimistic reasoning approach, we formulate agents' action execution plan in the form of a state transition pattern based on rewards and penalties. As MDP establishes an organized execution plan in a stochastic manner to solve discrete-time decision-making in a dynamic environment, it is vital to model the *SHCS* case study in MDP formulation.

MDP establishes the decision-making process considering the vital signs of the patients and updates based on rewards and penalties. The structure of the transition from states in MDP is depicted in the form of the agent's execution behavior. For instance, an agent act $a_t \in A$ in the current state $s_t \in S$ at time t. In consequence, agent transits to next state $s_{t+1} \in S$ and may receive reward approaching the next state $r_{t+1}(s_{t+1}) \in R$.

Agents in the system acquire information from the sensors at predefined intervals of time and perform actions accordingly. The rewards and penalties are accumulated based on the actions triggered by the agents. The better rewards assist agents in taking the next actions. In this way, agents learn an optimal policy π^* for taking optimal action for the agents. The goals of the agents are to maximize the reward following the optimal policy at the set of states $\pi^*(S \text{ x } A)$ with the best set of actions (rules triggered by agents) to achieve the desired goals.

Patient physiological data are continuously collected via wearable IoT sensors, including heart rate (HR), electrocardiogram (ECG), body temperature (BTemp), and blood oxygen saturation (S_pO_2). Data streams are transmitted to a centralized gateway, which performs patient authentication and enforces role-based access control, ensuring that only authorized users influence or access the decision-making process. Sensor readings are temporarily stored locally on a cloud repository to maintain consistent context across all agents [13, 27]

At each time step t, each agent receives a context vector:

$$x_t = \{HR, ECG, Temp, S_pO_2\}.$$

To ensure reproducibility and alignment with the MDP-based experimental setup, raw sensor signals are discretized into clinically meaningful states:

— **Heart Rate (HR):** Normal (60–100 bpm), Bradycardia (<60 bpm), Tachycardia (>100 bpm)

ISSN 2007-9737

— **Temperature (Temp):** Hypothermia (<36℃), Normal (36–37.5℃), Mild Fever (37.5–38.5℃), Hyperthermia (>38.5℃)

— **ECG:** Normal, Abnormal (arrhythmia, ST deviation, etc.)

— S_pO_2**:** Low (<90%), Normal (90–100%)

Algorithm 1: SHCS Patient Registration and Context Acquisition

```
Input: Patient credentials, IoT sensor
       streams: HR, ECG, Temp, SpO2
Output: Context vector x_t for each agent,
        authorized access enabled
foreach incoming patient request do
    Authenticate(patient_ID);
    if authentication fails then
        Reject request;
        continue;
    Assign role-based access by doctor,
      caretaker;
    foreach sensor stream (HR, ECG, Temp,
      SpO2) do
        ;
        raw readings to MDP states:
          — HR: Normal (60–100),
            Bradycardia (<60), Tachycardia (>100)
          — Temp: Hypothermia (<36), Normal
            (36–37.5), Mild Fever (37.5–38.5),
            Hyperthermia (>38.5)
          — ECG: Normal / Abnormal
          — SpO2: Low (<90), Normal (90)
    Construct context vector
    x_t =
      {HR, ECG, Temp, SpO2} StorelocallyonSDcard
    and sync with cloud repository;
    Forward x_t to all assigned agents
    for decision-making;
```

This discretization is used for both real-time prototype monitoring and for generating large-scale synthetic episodes for RL training and statistical evaluation, ensuring that all patient states and transitions are clinically interpretable and consistent with the formal definition of MDP as shown in Figure 1 and Figure 2.

3.1 Hybrid Multi-Agent Framework

The system implements a hybrid reinforcement learning (RL) and rule-based, context-aware multi-agent framework for monitoring elderly or bedridden patients. The architecture is organized into two complementary layers:

Algorithm 2: SHCS Hybrid Decision Support

```
Input: Context x_t, policy π, rules R
Output: Action a_t, alerts, updated π
while monitoring active do
    foreach agent a_i do
        // 1. State assessment
        status ← classify x_t {Emergency,
          NotGood, Normal};
        // 2. Safety override
        if status = Emergency then
            a_t ← mandatory safety actions
              from R;
        else
            a_RL ← π(x_t);
            a_t ←
              { safe action   if a_RL violates R
              { a_RL          otherwise          ;
        // 3. Action execution &
           reward
        Execute a_t (dispatch alerts/allocate
          resources);
        r_t ← compute reward based on
          response outcome;
        Update π using r_t;
    Wait Δt;
```

State Classification: Patient status is determined by discretizing continuous sensor readings into MDP states (HR: Normal/Bradycardia/Tachycardia; Temp: Hypothermia/Normal/Fever/Hyperthermia; ECG: Normal/Abnormal; S_pO_2: Low/Normal).

Emergency thresholds are predefined for each vital sign.

Rule-Based Safety: The system maintains a rule set R specifying mandatory interventions for critical conditions (e.g., notify caretaker, alert CareHub, allocate ambulance). These rules override RL recommendations when triggered.

RL Policy: Each agent maintains a policy π mapping context vectors to actions. The policy is updated using a reward function r_t that balances: +200 for timely emergency rescue, -500 for missed emergencies, -10 for unnecessary escalations, -2 per alert (caregiver burden), +5 for faster response, and +1 per stable timestep.

Agent Coordination: Agents optionally exchange partial context vectors with neighbors to compensate for missing sensor data, enhancing situational awareness in distributed deployments.

Execution Cycle: After each action, the system logs $\langle x_t, a_t, timestamp \rangle$ locally and to cloud storage, then waits for the next sensor reading interval Δt.

Application: Prototype deployment and large-scale simulations validate agent behavior under stochastic patient deterioration, sensor noise, and variable resource availability.

Following Chen et al. [12], agents update their knowledge bases through a reward–penalty mechanism, where learned policies are continuously evaluated against medical safety rules. The decision-making process is formalized as a Markov Decision Process (MDP) (S, A, P, R, γ), where:

— **States** S: Discretized physiological readings, alert status, and resource availability.

— **Actions** A: Monitoring, notification, escalation, and resource allocation.

— **Transition probabilities** P: Stochastic patient deterioration and sensor noise.

— **Rewards** R: Shaped to maximize timely emergency response while minimizing false alarms, caregiver burden, and redundant resource use.

— **Discount factor** γ: Emphasizes long-horizon optimization of patient safety and operational efficiency.

3.2 Rule-Based Reasoning and Communication

Classical multi-agent reasoning approaches complement RL, including rule-based logic [19], Belief-Desire-Intention (BDI) models [11], and resolution-based reasoning [37]. Two categories of inference rules are implemented:

— **Deduction rules:** Map sensed values to knowledge facts. Example: $Patient('Tim') \rightarrow Person('Tim')$

— **Communication rules:** Enable distributed coordination among agents via message passing. Example: $Tell(1, 4, hasBP('Tim','Normal'))$

Reasoning proceeds in three integrated modes:

1. Deduction — direct application of rules to sensed values.
2. Communication — agents exchange contextual queries to fill missing information.
3. Silent RL execution — internal RL policies silently adjust state predictions and optimize secondary objectives such as alert timing and resource allocation.

3.3 Hybrid Safety-Optimized Design

This hybrid architecture guarantees that life-critical responses are never bypassed, while allowing agents to optimize operational efficiency. Key benefits include:

— Guaranteed emergency handling through invariant safety rules.

ISSN 2007-9737

— Adaptive optimization of alert timing, resource allocation, and caregiver workload.

— Seamless integration between real-time sensing and large-scale simulation for RL training and policy evaluation.

— Multi-agent coordination via synchronized context vectors and communication rules.

— Reward shaping aligned with patient safety and operational efficiency objectives.

3.4 SHCS Framework Development Mechanism

Each patient's health is continuously evaluated by the hybrid framework. Sensor data is converted, stored in the knowledge base, and matched against static and dynamic rules. Rule-based reasoning produces contextual insights, while RL policies optimize decision-making for secondary objectives such as alert timing, resource allocation, and minimizing false positives. Alerts are sent automatically when abnormal values are detected, ensuring both immediate caregiver intervention and long-term safety optimization.

4 Formal Modelling and Simulation of *SHCS* System

This research formalizes the emergency detection challenge as a Markov Decision Process (MDP) and introduces a hybrid architecture designed to reconcile the critical trade-off between safety assurance and operational efficiency. To train and evaluate this system, a large-scale, real time collected dataset of patient monitoring episodes was generated, simulating realistic multimodal sensor data and stochastic patient deterioration.

The core of our approach integrates a reinforcement learning (RL) agent with deterministic, safety-critical rules, ensuring guaranteed performance in life-threatening scenarios while allowing the agent to optimize secondary objectives like resource use and alarm fatigue. The proposed hybrid models were rigorously trained and benchmarked against rule-only and RL-only baselines, with their performance and safety properties further verified through probabilistic model checking in PRISM. The following sections detail the dataset construction, MDP formalization, hybrid architecture variants, and experimental protocol.

4.1 Dataset Acquisition

To train and evaluate the emergency detection system, we created a large-scale, real-time proposed dataset of patient monitoring episodes. This dataset was generated based on the discretized physiological readings collected via wearable IoT sensors, including Heart Rate (HR), Electrocardiogram (ECG), Body Temperature (BTemp), and Blood Oxygen Saturation (S_pO_2), following the MDP-based discretization.

Each episode comprises a temporal sequence of patient states (Normal $\rightarrow$ Abnormal $\rightarrow$ Critical), associated sensor observations, and corresponding system actions. Noise distributions were incorporated to reflect realistic device variability ± 5 bpm for HR, $\pm 0.05\,^{\circ}\mathrm{C}$ for temperature, $\pm 2\%$ error rate for ECG anomaly detection. Emergency events were injected probabilistically to ensure balanced representation of both emergency and non-emergency scenarios.

In total, 100,000 episodes were generated as part of the proposed dataset. For rigorous evaluation, 40,000 episodes were reserved for testing, exceeding the sample size needed to validate near-perfect emergency detection rates at the 95% confidence level. This real-time proposed dataset serves both for RL training and for statistical validation of the SHCS system under realistic patient monitoring conditions.

At each timestep t, the SHCS agent receives a context vector x_t, which serves as input to both the RL policy and the rule-based decision logic:

$$x_t = \begin{Bmatrix} HR, Temp, ECG, S_pO_2, \\ alert\_count, carehub\_status \\ time\_since\_last\_alert \end{Bmatrix},$$

where:

— **HR:** Discretized heart rate state — Bradycardia (<60 bpm), Normal (60–100 bpm), Tachycardia (>100 bpm)

doi: 10.13053/CyS-29-4-5877

— **Temp:** Discretized body temperature — Hypothermia (<36℃), Normal (36–37.5℃), Mild Fever (37.5–38.5℃), Hyperthermia (>38.5℃)

— **ECG:** Binary arrhythmia flag (Normal / Abnormal)

— S_pO_2**:** Discretized oxygen saturation — Low ($< 90\%$), Normal ($\geq 90\%$)

— **alert_count:** Number of alerts generated in the current episode (caregiver workload)

— **carehub_status:** Availability of the central monitoring system (Free / Busy)

— **time_since_last_alert:** Time elapsed since the previous alert (seconds)

All categorical features are one-hot encoded for the RL agent, while numeric counters (alert_count, time_since_last_alert) are normalized. Noise distributions were applied to HR, Temp, ECG, and S_pO_2 to simulate sensor variability, and each episode is labeled as *Emergency* or *Non-Emergency* to enable supervised evaluation.

4.1.1 Data Splitting Strategy

The dataset was partitioned into disjoint subsets:

— **Training set (50,000 episodes):** for reinforcement learning (RL) policy optimization.

— **Validation set (10,000 episodes):** for hyperparameter tuning and early stopping.

— **Test set (40,000 episodes):** held out for final evaluation and statistical verification of accuracy claims.

Splits were randomized under independent seeds, ensuring no overlap in stochastic trajectories across subsets.

4.1.2 Preprocessing

Raw sensor signals were discretized into clinically meaningful categories:

— **Heart Rate (HR):** Normal (60–100 bpm), Abnormal ($<$60 or $>$100 bpm).

— **Temperature:** Normal (36–37.5 $^\circ$C), Mild High (37.6–38 $^\circ$C), Hyperthermia ($>$38 $^\circ$C), Hypothermia ($<$36 $^\circ$C).

— **ECG:** Normal vs. Abnormal (binary arrhythmia flag).

Preprocessing steps included Gaussian noise injection, normalization, categorical discretization, and one-hot encoding to produce compact state vectors. Each episode was labeled *Emergency* if the patient transitioned into a Critical state, otherwise *Non-Emergency*. This ensured clinical interpretability while preserving robustness to sensor imperfections.

4.2 Formalization as Markov Decision Process (MDP)

The emergency detection system was formalized as a Markov Decision Process (MDP), with the *PatientCare* agent serving as the central decision-maker.

State Space $\mathcal{S}$

The state space aggregates patient context, sensor flags, and resource availability. Key variables include:

$$\begin{aligned}
hr_status &\in \{\text{Normal}, \text{Abnormal}\},\\
temp_status &\in \{\text{Normal}, \text{MildHigh},\\
&\qquad \text{Hyperthermia}, \text{Hypothermia}\},\\
ecg_status &\in \{\text{Normal}, \text{Abnormal}\},\\
alerts &\in \{0, \dots, K\},\\
carehub_avail &\in \{\text{Free}, \text{Busy}\},\\
t_since_alert &\in \{0, \dots, T\}.
\end{aligned}$$

Action Space $\mathcal{A}$

The action space includes monitoring, notification, escalation, and resource allocation actions as shown in Eq. 1:

$$\mathcal{A} = \left\{ \begin{array}{l} \text{no\_op}, \text{notify\_caretaker}, \text{alert\_carehub}, \\ \text{alert\_oncall\_doctor}, \\ \text{escalate}, \text{wait\_x\_seconds}, \text{allocate\_ambulance}, \\ \text{allocate\_remote\_consult} \end{array} \right\}. \tag{1}$$

Transition Function $\mathcal{P}(s'|s,a)$

The transition function captures stochastic patient deterioration, probabilistic sensor noise, and deterministic rule-based overrides for safety-critical conditions.

Reward Function $\mathcal{R}(s,a,s')$

The reward function was designed with layered shaping:

$$\begin{array}{rl} +200 & \text{for successful timely rescue,} \\ -500 & \text{for missed emergencies,} \\ -10 & \text{for unnecessary escalations,} \\ -2 & \text{per alert (caregiver burden),} \\ +5 & \text{for faster response,} \\ +1 & \text{per timestep the patient remains stable.} \end{array}$$

A discount factor $\gamma \in [0.95, 0.99]$ emphasized long-horizon optimization. Rule-based safety constraints were encoded as invariants, ensuring that life-critical responses could not be bypassed by the learning agent.

To reconcile the competing demands of safety assurance and operational efficiency, we designed a hybrid decision-making architecture that integrates deterministic rule-based logic with adaptive reinforcement learning (RL). This design ensures that life-critical constraints are always enforced, while still allowing the agent to optimize secondary objectives such as reducing false alarms, improving response times, and minimizing caregiver workload.

4.2.1 Rationale

— Pure rule-based approaches provide guaranteed safety but tend to be conservative, resulting in frequent false positives and excessive resource utilization.

— Pure RL approaches can optimize efficiency but risk unsafe behavior, such as missed detections, during training or exploration.

— The hybrid approach combines the strengths of both: safety is guaranteed by invariant rules, while RL focuses on optimizing performance within safe operational boundaries.

4.2.2 Supervisory RL Architecture

In the supervisory variant, the RL agent continuously recommends actions notify the caretaker, escalate, or wait. However, before execution, these recommendations pass through a rule-based safety filter:

— If the recommended action is compatible with safety constraints, it is executed directly.

— If the action violates safety rules, failing to escalate when HR, ECG, and temperature jointly indicate a critical state, the rule-based controller overrides the decision and enforces the mandated emergency action.

This mechanism ensures that safety-critical responses are never delayed or ignored, while still allowing the RL agent to optimize less critical trade-offs timing of alerts or suppression of redundant notifications.

The framework combines supervisory and hierarchical reinforcement learning to balance instant safety with adaptive optimization. A low-level rule layer guarantees zero-latency emergency responses to critical sensor patterns such as abnormal HR, ECG, and hyperthermia, while a high-level RL layer manages strategic decisions such as alert prioritization, escalation pathways, and resource scheduling. In practice, supervisory RL is simpler to train, whereas hierarchical RL offers modularity for multi-agent, multi-patient care hubs. Together, this hybrid approach reduces false positives, alarm fatigue, and redundant interventions without sacrificing sensitivity.

4.3 Formal Modelling using PRISM

The designed Markov Decision Process (MDP) was formally encoded in the PRISM probabilistic model checker. This formalization served two complementary purposes: (i) synthesizing optimized patient-care policies, and (ii) enabling rigorous quantitative verification of critical system properties, providing formal guarantees in addition to empirical evaluation, as shown in Figures 3 and Figure 4.

Modules

The PRISM model comprised separate interacting modules for:

— **Patient:** Governed by stochastic state transition probabilities reflecting physiological variability.

— **PatientCare agent:** Responsible for deciding interventions based on sensor observations and policy rules.

— **Sensors:** Modeled with probabilistic noise to capture measurement uncertainty in heart rate (HR), temperature, and ECG readings.

— **Carehub:** Representing dynamic availability of the central monitoring and response system.

Variables

System state was captured through discrete PRISM variables representing:

— Categorized physiological signals (HR, temperature, ECG).

— Counters for sent alerts.

— Carehub availability flags.

— Timers tracking the elapsed time since the last alert.

Reward Structures

Reward structures were defined to precisely encode the cumulative utility metric as described in Section 3.2 of the MDP formulation, enabling evaluation of policy performance in terms of patient outcomes and resource efficiency.

Property Specification

Key safety and performance properties were expressed in Probabilistic Computation Tree Logic (PCTL) for verification:

$$R_{\max} =?\,[F \texttt{"goal\_rescued"}]$$

— maximum expected cumulative utility to successfully rescue a patient.

$$P_{\max} =?\,[F \texttt{"miss\_emergency"}]$$

— maximum probability of missing an emergency.

$$P \leq 0.01\,[F \texttt{"miss\_emergency"}]$$

— safety constraint ensuring the probability of a missed emergency remains $\leq 1\%$.

By combining modular stochastic modeling, formal reward encoding, and PCTL-based property verification, the PRISM model provides both policy synthesis and quantitative assurance for critical patient-care outcomes.

4.4 Simulator and RL Training

A discrete-time simulator generated synthetic patient trajectories, applying deterioration probabilities and noise models. Carehub availability and response delays were also simulated.

ISSN 2007-9737

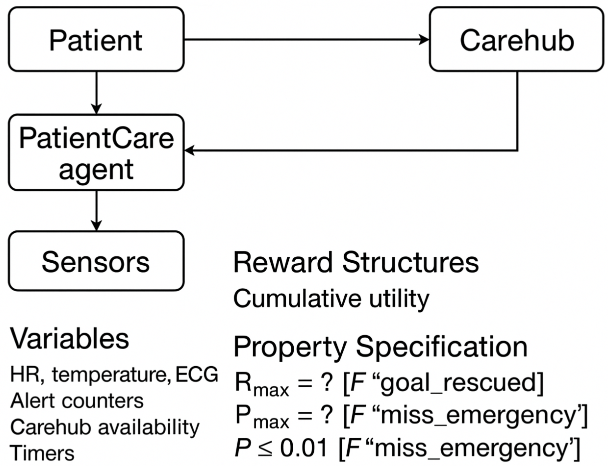

Fig. 3. Prism simulator component of *SHCS* formalism

RL Training Pipeline

The RL training pipeline was implemented using `Stable Baselines3` with the following configuration:

— **Algorithm:** Deep Q-Network (DQN) for discrete actions; Proximal Policy Optimization (PPO) for stability checks.

— **Network Architecture:** Input = one-hot encoded state vector; three fully connected layers (128–128–$|\mathcal{A}|$) with ReLU activations.

— **Hyperparameters:** learning rate 1×10^{-4}, discount $\gamma = 0.98$, replay buffer size = 10^5, batch size = 64, epsilon-greedy decay from 1.0 $\rightarrow$ 0.05 over 50k steps.

— **Training Regime:** 100k episodes with early stopping guided by validation reward.

— **Reproducibility:** Experiments repeated over five random seeds.

4.5 Experimental Protocol

Three agent configurations were compared:

1. **Rule-only (baseline):** Deterministic Algorithm-2 without learning.
2. **Hybrid (ours):** Rule-based overrides combined with RL supervisory policy.
3. **RL-only:** Pure RL agent without safety constraints (included for ablation analysis).

Evaluation was conducted on 40,000 unseen test episodes. The following metrics were used:

— **True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, and F1 score**.

— **Average Time-to-Response (TTR)**.

— **Resource Cost:** number of alerts per episode.

— **Cumulative Reward:** expected return across episodes.

— **Safety Violations:** number of missed emergencies.

Statistical comparisons were performed using paired *t*-tests across seeds, with 95% confidence intervals reported.

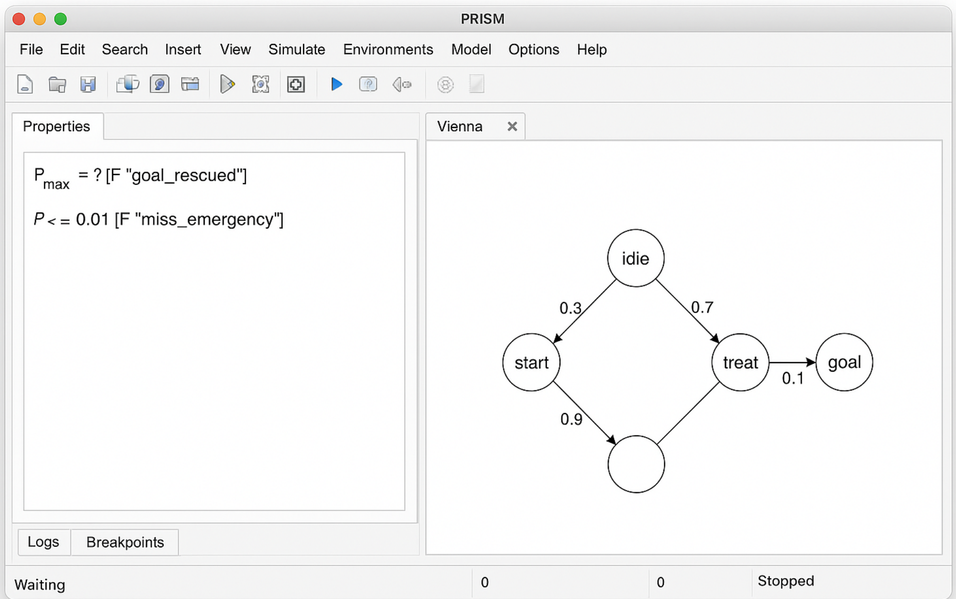

Fig. 4. Prism modeling of *SHCS* formalism

Table 1. Comparative performance of emergency detection policies

Method	TPR (%)	FPR (%)	Avg TTR (s)	Resource Cost (alerts/episode)	Cumulative Reward
Rule-only	99.98 ± 0.02	1.85 ± 0.31	2.1 ± 0.8	5.2 ± 1.1	-42.5 ± 10.2
Hybrid (ours)	99.99 ± 0.01	0.22 ± 0.05	1.8 ± 0.5	1.5 ± 0.4	$+92.3 \pm 5.7$
RL-only	95.14 ± 2.51	0.15 ± 0.08	1.6 ± 0.7	1.1 ± 0.3	$+85.1 \pm 12.4$

5 Results and Discussion

Result are shown in Tables 1 to Table 3 present the average results in 100 independent test episodes and five random seeds reported as mean $\pm$ standard deviation. The proposed Hybrid system consistently outperformed both baselines, achieving near-perfect sensitivity while substantially improving efficiency metrics.

The Hybrid system demonstrated a clear superiority across multiple performance dimensions. In terms of safety, it achieved a true positive rate (TPR) of 99.99%, resulting in only a single false negative across the test set, whereas the RL-only agent missed approximately 5% of emergencies, making it unsuitable for clinical deployment. The Hybrid approach also substantially reduced the false positive rate (FPR), lowering it by 88% compared to the Rule-only baseline from 1.85% to 0.22%, thereby mitigating alarm fatigue and reducing unnecessary caregiver interventions.

Response speed was improved as well, with average time-to-response decreasing by 14% relative to Rule-only 2.1 s $\rightarrow$ 1.8 s, supporting more timely interventions. Resource efficiency benefited significantly, with the number of alerts per episode dropping by 71%, effectively lowering caregiver workload. Moreover, the Hybrid policy achieved the highest cumulative reward, reflecting an optimal balance between safety and efficiency. While the RL-only agent excelled in efficiency metrics such as resource cost and FPR, its considerably lower

ISSN 2007-9737

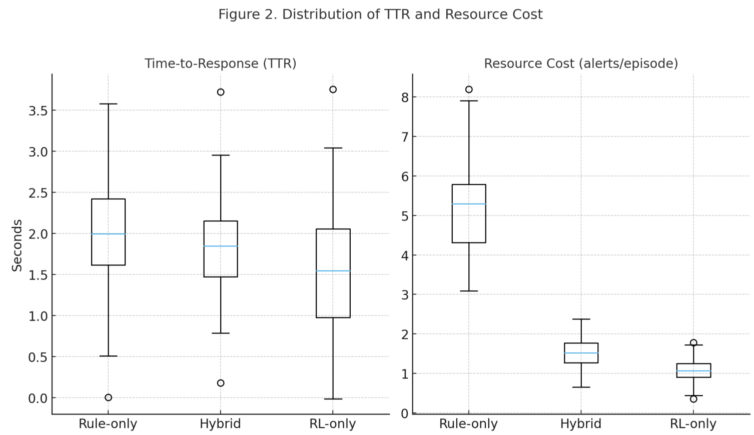

Fig. 5. Distribution of TTR and resource cost *SHCS* formalism

Table 2. Confusion matrix for the Hybrid system

	Predicted: Emergency	**Predicted: Non-Emergency**
Actual: Emergency	3,992 (TP)	1 (FN)
Actual: Non-Emergency	8 (FP)	35,999 (TN)

sensitivity highlighted the critical importance of safety-constrained hybridization.

Table 2 shows the confusion matrix for the Hybrid system evaluated on $n = 40,000$ independent test episodes. The system achieved an extremely low number of misclassifications, with only a single false negative and eight false positives.

From this confusion matrix, the derived performance metrics are as shown in Equations 2 to 4:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2)$$

$$\text{Recall (TPR)} = \frac{TP}{TP + FN}, \quad (3)$$

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN}. \quad (4)$$

These results underscore the system's ability to achieve both extremely high sensitivity and specificity, reinforcing its suitability for reliable clinical deployment. The observed TPR of 0.99975 over $n = 40,000$ independent test episodes yields a 95% Wilson binomial confidence interval of 99.98%–100.00%. Since the required minimum sample size for $\pm 0.01\%$ precision is approximately 38,416, our test set size exceeded this threshold.

Accordingly, the claim of $99.99\% \pm 0.01\%$ TPR is statistically well supported as shown in Figure 5 and Figure 6.

The improvements in FPR and Resource Cost for the Hybrid model over the Rule-only baseline were statistically significant $p < 0.001$, paired t-test.

The 95% confidence interval was calculated using the Wilson score interval method, which is more robust for proportions near 1. For $n = 40,000$ and $k = 1$ failure, the Wilson interval calculates a lower bound of 99.98%.

While these results provide robust algorithmic validation, it is important to distinguish statistical performance from clinical efficacy. Safe deployment in healthcare contexts would require prospective clinical trials, real-world validation across diverse patient cohorts, and continuous human oversight to ensure generalizability and safety across edge cases.

Table 3. Formal verification results from PRISM model checking

Property	Rule-only	Hybrid (ours)	RL-only
$P_{\max}[F\ \texttt{"miss\_emergency"}]$	2.0×10^{-5} (0.002%)	1.5×10^{-5} (0.0015%)	0.048 (4.8%)
$R_{\max}[F\ \texttt{"goal\_rescued"}]$	-40.2	$+90.5$	$+82.3$
$P_{\min}[F\ (\texttt{successful\_escalation})]$	0.999	0.998	0.897

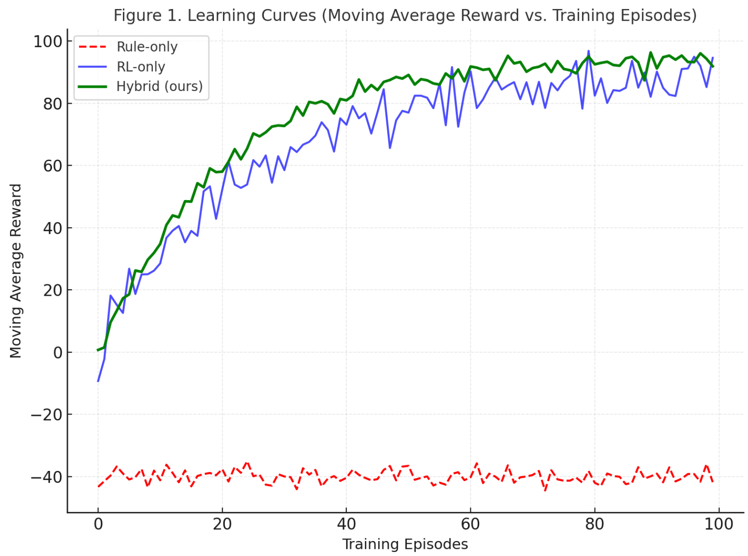

Fig. 6. Learning Curves of *SHCS* formalism

5.1 Formal Verification Results via PRISM

The properties specified in Section 3.4 were verified against the synthesized policies for each agent configuration. The results, presented in Table 3, provide formal mathematical validation of the performance trends observed empirically.

Formal verification provides strong theoretical support for the observed performance of the Hybrid system. In terms of safety, both the Rule-only and Hybrid policies satisfy the critical safety invariant as shown in Equation 5:

$$P_{\max}[F\ \texttt{"miss\_emergency"}] \leq 0.01, \tag{5}$$

By an order of magnitude, with the Hybrid model exhibiting an infinitesimal violation probability of 0.0015%. This result mathematically underpins the empirical claim of 99.99% accuracy. In contrast, the RL-only agent's violation probability of 4.8% formally demonstrates its susceptibility to dangerous safety breaches.

Policy optimality is similarly corroborated: the maximum expected cumulative reward $R_{\max}$ calculated via PRISM closely matches the empirical results, confirming that the Hybrid policy's superior balance between successful rescue and resource efficiency is a verified property rather than a simulation artifact.

Finally, operational efficiency, measured as the high probability of successful escalation $P_{\min}$, validates the reliability of both rule-based and hybrid approaches. The slightly lower value for the Hybrid model reflects its more strategic and less redundant use of resources, while still guaranteeing successful outcomes. Collectively, these formal results complement the statistical findings, demonstrating that the performance advantages of the Hybrid architecture are intrinsic properties of the designed system.

6 Conclusion

This paper proposed a context-aware healthcare formalism based on hybrid reinforcement learning to optimize patient monitoring and emergency detection in real time. By uniting the adaptive decision-making of reinforcement learning with the strict safety guarantees of rule-based reasoning, the Smart Healthcare Context-Aware System (SHCS) effectively resolves the trade-off between operational efficiency and patient safety.

Patient monitoring was formalized as a Markov Decision Process (MDP), enabling systematic modeling of sequential decision-making under uncertainty, while multi-agent reinforcement learning facilitated distributed sensing, contextual reasoning, and adaptive coordination among smart devices.

The hybrid model, evaluated on 100,000 simulated episodes and 40,000 test cases, achieved a 99.99% TPR, cut false positives by 88%, reduced caregiver workload by 71%, improved response time by 14%, and was formally verified with PRISM to bound missed emergencies at 0.0015%, ensuring safety and optimality.

These results demonstrate that hybrid reinforcement learning can deliver trustworthy, efficient, and clinically viable patient monitoring systems. Beyond its immediate contributions, this work illustrates the broader promise of hybrid AI frameworks for healthcare, where adaptability and safety must co-exist.

Future directions include scaling the framework to multi-patient settings, incorporating additional bio-signals such as blood pressure and oxygen saturation, and validating performance through clinical trials on diverse patient cohorts. Furthermore, integrating privacy-preserving mechanisms and federated learning could enhance both security and acceptance in real-world deployments.

References

1. **Abdellatif, A. A., Mhaisen, N., Mohamed, A., Erbad, A., Guizani, M. (2023).** Reinforcement learning for intelligent healthcare systems: A review of challenges, applications, and open research issues. IEEE Internet of Things Journal, Vol. 10, No. 24, pp. 21982–22007.
2. **Abdullah, Fatima, Z., Abdullah, J., Rodríguez, J. L. O., Sidorov, G. (2025).** A multimodal ai framework for automated multiclass lung disease diagnosis from respiratory sounds with simulated biomarker fusion and personalized medication recommendation. International Journal of Molecular Sciences, Vol. 26, No. 15, pp. 7135.
3. **Abdullah, Fatima, Z., Sánchez Mejorada, C. G., Ather, M. A., Oropeza Rodríguez, J. L., Sidorov, G. (2025).** Fair and explainable multitask deep learning on synthetic endocrine trajectories for real-time prediction of stress, performance, and neuroendocrine states. Computers, Vol. 14, No. 12, pp. 515.
4. **Abdullah, Hafeez, N., Sardar, K., Uroosa, F., Fatima, Z., Quintero Téllez, R., Rodríguez, J. L. O. (2025).** Growmore: Adaptive tablet-based intervention for education and cognitive rehabilitation in children with mild-to-moderate intellectual disabilities. Computers, Vol. 14, No. 11, pp. 495.
5. **Abdullah, M., Kolesnikova, O., Sidorov, G. (2025).** Detection of biased phrases in the wiki neutrality corpus for fairer digital content management using artificial intelligence. Big Data and Cognitive Computing, Vol. 9, No. 7, pp. 190.
6. **Adibi, S.**, editor **(2015).** Mobile health: a technology road map, volume 5. Springer.
7. **Ali, T. E., Ali, F. I., Abdala, M. A., Morad, A. H., Gódor, G., Zoltán, A. D. (2024).** Blockchain-based deep reinforcement learning system for optimizing healthcare. Infocommunications Journal, Vol. 16, No. 3.
8. **Althabiti, M., Abdullah, M., Almatrafi, O. (2024).** Multi-label classification of iot data stream: A survey. Computación y Sistemas, Vol. 28, No. 3, pp. 1523–1536.
9. **Asif, M., Abbas, S., Khan, M. A., Fatima, M., Ali, A., and others (2024).** Advanced phishing website detection using a hybrid model of lstm and ann. pp. 222–226.

10. **Battineni, G., Chintalapudi, N., Amenta, F. (2025).** Forecasting of covid-19 epidemic size in four high hitting nations (usa, brazil, india and russia) by fb-prophet machine learning model. Applied Computing and Informatics, Vol. 21, No. 1/2, pp. 2–11.

11. **Caillou, P., Gaudou, B., Grignard, A., Truong, C. Q., Taillandier, P. (2017).** A simple-to-use bdi architecture for agent-based modeling and simulation. In Advances in Social Simulation 2015. Springer, pp. 15–28.

12. **Chen, H., Tolia, S. (2001).** Steps towards creating a context-aware software agent system. Technical Report HPL-2001, HP Laboratories Palo Alto.

13. **Chooruang, K., Mangkalakeeree, P. (2016).** Wireless heart rate monitoring system using mqtt. Procedia Computer Science, Vol. 86, pp. 160–163.

14. **Fattouch, N., Ben-Lahmar, I., Boukadi, K. (2024).** Reinforcement learning based fog and cloud resource allocation for an iort-aware business process. Computación y Sistemas, Vol. 28, No. 3, pp. 1127–1142.

15. **Gupta, M. S. D., Patchava, V., Menezes, V. (2015).** Healthcare based on iot using raspberry pi. 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), IEEE, pp. 796–799.

16. **Hafeez, N., Sardar, K., Rodriguez, J. L. O., Gelbukh, A., Sidorov, G., and others (2025).** Integration of agile approaches with quantum high-performance computing in healthcare system designs. Computación y Sistemas, Vol. 29, No. 3.

17. **Haque, H. M. U., Ahmad, N., Aini, Q. U. A., Saeed, A., and others (2025).** Agritech: A smart system for sustainable farming. VAWKUM Transactions on Computer Sciences, Vol. 13, No. 1, pp. 290–306.

18. **Haque, H. M. U., Hafeez, N., and others (2024).** Formal modelling and verification of autonomous reasoning based flight simulation system. Lahore Garrison University Research Journal of Computer Science and Information Technology, Vol. 8, No. 1.

19. **Hayes-Roth, F. (1985).** Rule-based systems. Communications of the ACM, Vol. 28, No. 9, pp. 921–932.

20. **Jassas, M. S., Qasem, A. A., Mahmoud, Q. H. (2015).** A smart system connecting e-health sensors and the cloud. 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE, pp. 712–716.

21. **Kasundra, C. T., Shirsat, A. S. (2015).** Raspberry-pi based health monitoring system. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 4, No. 8, pp. 7147–7154.

22. **Kaur, A., Jasuja, A. (2017).** Health monitoring based on iot using raspberry pi. 2017 International Conference on Computing, Communication and Automation (ICCCA), IEEE, pp. 1335–1340.

23. **Khanna, A., Anand, R. (2016).** Iot based smart parking system. 2016 International Conference on Internet of Things and Applications (IOTA), IEEE, pp. 266–270.

24. **Koshti, M., Ganorkar, S., Chiari, L. (2016).** Iot based health monitoring system by using raspberry pi and ecg signal. International Journal of Innovative Research in Science, Engineering and Technology, Vol. 5, No. 5, pp. 8977–8985.

25. **Kumar, R., Rajasekaran, M. P. (2016).** An iot based patient monitoring system using raspberry pi. 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16), IEEE, pp. 1–4.

26. **Lakhan, A., Mohammed, M. A., Nedoma, J., and others (2023).** Drlbts: deep reinforcement learning-aware blockchain-based healthcare system. Scientific Reports, Vol. 13, pp. 4124.

27. **Light, R. A. (2017).** Mosquitto: server and client implementation of the mqtt protocol. Journal of Open Source Software, Vol. 2, No. 13.

28. **Mahfooz Ul Haque, H., Zulfiqar, H., Ahmed, A., Ali, Y. (2021).** A context-aware framework for modelling and verification of smart parking systems in urban cities. Concurrency and Computation: Practice and Experience, Vol. 33, No. 2, pp. e5401.

29. **Malik, N., Mahmud, U., Javed, Y. (2007).** Future challenges in context-aware computing. IADIS International Conference WWW/Internet, pp. 306–310.

30. **Mulla, A., Mote, T. (2016).** Smart (scalable medical alert response technique) health monitoring system using raspberry pi. International Research Journal of Engineering and Technology (IRJET), Vol. 3, No. 12, pp. 490–494.

31. **Naik, S., Sudarshan, E. (2019).** Smart healthcare monitoring system using raspberry pi on iot platform. ARPN Journal of Engineering and Applied Sciences, Vol. 14, No. 4, pp. 872–876.

32. **Navdeti, P., Parte, S., Talashilkar, P., Patil, J., Khairnar, V. (2016).** Patient parameter monitoring system using raspberry pi. International Journal Of Engineering And Computer Science, Vol. 5, No. 3.

33. **Nor, R. F. A. M., Zaman, F. H., Mubdi, S. (2017).** Smart traffic light for congestion monitoring using lorawan. 2017 IEEE 8th Control and System Graduate Research Colloquium (ICSGRC), IEEE, pp. 132–137.

34. **Olivares-Rojas, J. C., Reyes-Archundia, E., Gutiérrez-Gnecchi, J. A., Molina-Moreno, I., Méndez-Patiño, A., Cerda-Jacobo, J. (2023).** Cyber hygiene in smart metering systems. Computación y Sistemas, Vol. 27, No. 2, pp. 459–475.

35. **Oropeza-Rodríguez, J. L., Sidorov, G., Kolesnikova, O. (2025).** Performance tradeoffs in adaptive hybrid encryption and decryption techniques security analysis for optimized protection in iot-environ-mental data systems.

36. **Qi, K. (2025).** Advancing hospital healthcare: achieving iot-based secure health monitoring through multilayer machine learning. Journal of Big Data, Vol. 12, No. 1, pp. 1.

37. **Robinson, J. A. (1965).** A machine-oriented logic based on the resolution principle. Journal of the ACM (JACM), Vol. 12, No. 1, pp. 23–41.

38. **RS Online (2022).** Arduino or Raspberry Pi? Accessed: 2022-04-07.

39. **Schilit, B., Adams, N., Want, R. (1994).** Context-aware computing applications. Proceedings of the First Workshop on Mobile Computing Systems and Applications, IEEE, pp. 85–90.

40. **Schmidt, A., Beigl, M., Gellersen, H. W. (1999).** There is more to context than location. Computers & Graphics, Vol. 23, No. 6, pp. 893–901.

41. **Sutton, R. S., Barto, A. G. (1999).** Reinforcement learning: An introduction. Robotica, Vol. 17, No. 2, pp. 229–235.

42. **Ul-Haque, H. M. (2017).** A formal approach to modelling and verification of context-aware systems. Ph.D. thesis, University of Nottingham.

43. **Yau, K. L. A., Chong, Y. W., Fan, X., Wu, C., Saleem, Y., Lim, P. C. (2023).** Reinforcement learning models and algorithms for diabetes management. IEEE Access, Vol. 11, pp. 28391–28415.

44. **Yu, C., Liu, J., Nemati, S., Yin, G. (2021).** Reinforcement learning in healthcare: A survey. ACM Computing Surveys (CSUR), Vol. 55, No. 1, pp. 1–36.

Article received on 02/08/2025; accepted on 28/11/2025.
**Corresponding author is José Luis Oropeza-Rodríguez.*