DOI: 10.5281/zenodo.18042162

Adaptive Runtime Validation of a Responsive Neurostimulation Device in a Body Area Network Using Hierarchical Colored Timed Petri Nets and Hierarchical Reinforcement Learning

Safaa Majid Fakhry Al-Sherify[1], Negar Majma[2&3*], Asaad Noori Hashim Al-Shareefi[4] and Zohreh Fotouhi[5]

[1]Department of Computer Engineering, Isf.C., Islamic Azad University, Isfahan, Iran
Email: Safa.majid@iau.ac.ir; https://orcid.org/0009-0005-9890-1572
[2]Department of Computer Engineering, Isf.C., Islamic Azad University, Isfahan, Iran
[3]Department of Computer Engineering, Naghshejahan Higher Education Institute, Isfahan, Iran
Email: Negar.majma@iau.ir
https://orcid.org/0000-0002-9774-1248
[4]Computer Science Department and Mathematics, University of Kufa, Najaf, Iraq
Email: asaad.alshareefi@uokufa.edu.iq; https://orcid.org/0000-0002-4592-3218
[5]Department of Computer Engineering, Isf.C., Islamic Azad University, Isfahan, Iran.
Email: z.fotouhi@iau.ac.ir; https://orcid.org/0000-0002-6021-0849

*Corresponding author: Negar.majma@iau.ir

Published: 24 December 2025
Accepted: 19 November 2025
Received: 08 July 2025

Abstract: Recent advances in computing, networking, and medical sensing technologies have enabled the design of brain neural stimulation devices for the detection, monitoring, and treatment of epileptic seizures. However, online validation of these devices under critical conditions remains a major challenge from both safety and dependability perspectives. In this paper, we propose an adaptive runtime validation framework for a brain neural stimulation device that combines Hieratical Timed Colored Petri Nets (HTCPN), fuzzy logic, and hierarchical reinforcement learning. The execution structure of the device and the transitions between brain states are formalized using a HTCPN model, while a fuzzy system maps continuous brain and physiological measurements to four discrete risk levels (S1–S3) corresponding to normal activity, probable epileptic activity, and seizure onset which are then used by the hierarchical reinforcement learning agent. On top of this model, we design a two-layer reinforcement learning agent: at the upper level, a Manager uses the fuzzy output and physiological sensors (heart rate, temperature, and blood pressure) to choose among three actions—no intervention, delegating the decision to the stimulation layer (Worker), or inhibiting stimulation—while at the lower level, the Worker generates the stimulation pattern whenever it is activated. The reward function is defined based on driving physiological variables toward safe ranges and on agreement or disagreement with expert decisions. For training, starting from 10 expert-defined reference scenarios, approximately 1,500 augmented scenarios are generated by adding Gaussian noise to brain and physiological signals, systematically varying physiological profiles, and interpolating between same-class scenarios; the hierarchical agent is then trained on this augmented set and subsequently evaluated on the original 10 reference scenarios. Simulation results show that the proposed hierarchical agent learns a stable policy that, in critical scenarios, delegates stimulation control to the Worker and achieves positive and clinically meaningful cumulative rewards. A comparison with a simple Q-learning baseline demonstrates that embedding a hierarchical structure on top of the TCPN–fuzzy model leads to more structured and effective decision-making for adaptive runtime validation of the neural stimulation device within a Body Area Network environment.

Keywords: *Responsive Neurostimulation (RNS) device, Body Area Network (BAN), Hierarchical Timed Colored Petri nets (HTCPN), Fuzzy logic, Hierarchical reinforcement learning, Data augmentation for medical scenarios*

1. **Introduction**

Today, the use of medical devices for treatment, diagnosis, and improving human health status is expanding and developing. A medical device is officially defined by the World Health Organization (WHO) as "any type of instrument, apparatus, implement, machine, appliance, implant, reagent for in vitro use, software, or other similar or related article which is intended by the manufacturer to be used alone or in combination with other devices for specific medical purposes in humans" [1]. Examples of these implantable and wearable medical devices include heart pacemakers, defibrillators, insulin delivery pumps and glucose monitoring, deep brain stimulation, intrathecal drug delivery, and many other devices that are used for diagnostic, monitoring, and therapeutic functions [2]. New methods and technologies for reducing the size of these devices are expanding, which leads to these devices getting closer to the human body without even the slightest discomfort for the user or creating danger for them. For example, an individual can have a medical device in the form of a wireless body area sensor network (WBAN) with internal sensors in their body that calculates their heart rate at any time and sends this data via the internet to the doctor. WBANs not only obtain fixed information about patients and their body parameters but can also detect diseases and facilitate continuous medical treatments and interventions.

Accordingly, personal healthcare systems based on implantable and wearable medical devices (IWMDs) are raised, which are used to enhance the quality of diagnosis and treatment of a wide range of medical conditions. IWMDs typically include wireless communications through which they can connect to internal and external diagnostic and control equipment or to body area networks (BANs) to create a personal healthcare system (PHSs). To form a PHS, it is usually necessary to collect physiological data from the human body using sensors, and this data is delivered to remote controllers. These controllers use the data for analyzing the patient's condition and decision-making. The controllers can be a software agent that autonomously and intelligently makes decisions and acts.

As IWMD devices become smarter, their reliability decreases and it becomes less possible to assure their correct performance. These devices, with increasing intelligence, are more susceptible to attacks and intrusions. Various problems, including hardware failures, software errors, wireless attacks, malware and software abuses, and side-channel attacks, can undermine the performance of IWMD and BAN [3]. Solutions such as fault tolerance, information security, redundancy, and encryption can increase assurance and trust in the functioning of medical devices. The FDA organization between 2006 and 2011 registered 5294 recalls for medical devices. Of the 5294, 1210 recalls were related to computer system problems in these devices, which 90.5% were related to Class II devices such as ECG. Most Class II recalls (80%) were related to software and hardware. The remaining recalls were related to batteries, input/output devices, and other components [4].

One of the validation methods for a medical device is examining its behavior in comparison with the correct and specified performance and determining whether the device's behavior is compatible with its actual performance in the real world and meets patients' needs or not. To achieve this goal, the use of accurate and explicit models that can simulate the complexities of the device and its behaviors in various conditions is essential. One of the powerful tools in this field is Petri nets, which, especially in more advanced versions such as Colored Petri Nets (CPN), provide unparalleled capabilities for modeling and analyzing the behavior of complex devices. The main reasons for using this type of networks in medical device validation can include 1) graphical and comprehensible representation of the system, 2) capability to model complex and multi-part systems, 3) management of complex data using colors, 4) analysis of temporal system behavior, 5) adaptability to uncertain behaviors, 6) analysis and evaluation of system safety, 7) capability for comparison and compliance, 8) strong analysis and networking tools, and 9) extensibility for interaction with intelligent systems.

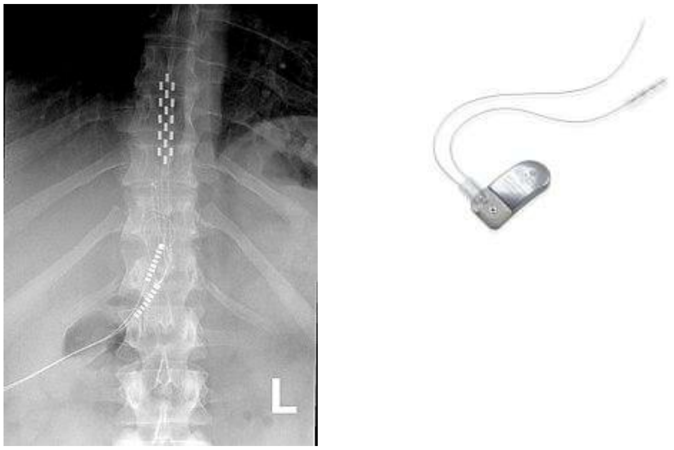
Runtime validation of medical devices, due to the importance of device execution, has a significant runtime challenge. Among the advantages that this runtime validation has, the following can be mentioned: (1) discovery of instantaneous errors or unexpected device behavior; (2) high flexibility due to the presence of an intelligent agent for responding to various conditions; (3) enhancement of device execution safety that can occur from incorrect device performance; (4) dynamic management of the device due to the Petri net-based knowledge base that dynamically improves with changes in patient behavior through reinforcement learning. Therefore, in this article, using brain sensors, the executive structure of the brain neural stimulation device as an implantable device in the body is modeled in Petri networks, and by employing body physiological sensors as learning data, adaptive and modified decision-making is performed at runtime. Considering the fuzzy and uncertain status of the body, sensor values and executive rules are simulated fuzzily, which are used in the executive conditions of the applied Petri network. This modelling can support runtime decision-making, reduce device errors, and provide continuous runtime validation of the RNS device.

In the following, first in section 2, the research literature is reviewed, and then in section 3, the research background is examined. In section 4, the proposed model is described, and in section 5, the presented execution scenarios are provided. Finally, in sections 6 and 7, respectively, the evaluation of the proposed solution and the conclusion are addressed.

2. Literature Review

2.1. Responsive Neural Stimulation Device

Responsive Neural Stimulation (RNS) is an implantable device used for the treatment of epilepsy. This device is an intelligent, adjustable, and reversible device where the implantation site and method of use are specifically customized for each individual. This device learns the individual's brain activities, and its settings can be changed individually for each patient [5]. This device detects and responds to abnormal electrical activity in the brain by automatically sending small electrical pulses to the area where the seizure starts to prevent the occurrence of seizures or reduce the number of seizures. This device does not cure epilepsy but significantly improves the condition and quality of life of patients by reducing the number and severity of seizures. The use of this medical device is particularly suitable for individuals for whom anti-seizure medications have not been effective or other surgical options are not suitable for them. The US Food and Drug Administration has approved this device for the treatment of epileptic seizures and conditions associated with chronic pain. An example of this device can be seen in Figure 1a, and the location of the electrodes of this device in Figure 1b.



(b) (a)

Figure 1 - An example of the RNS device and its installation location

The RNS system is similar to a pacemaker device. This system monitors brain waves and responds if unusual activity or seizure-like patterns are identified. This system can send small pulses or short bursts of electrical stimulation to the brain to prevent the onset of a seizure or to prevent its spread from a focal seizure to a more widespread one. Individuals do not feel the stimulation; this process does not cause pain or abnormal sensation. The RNS device has three main operational phases in which the device automatically monitors and responds to the patient's brain activity. These phases are:

1. Monitoring Phase: In this phase, the device continuously monitors the brain's electrical activity through electrodes placed in specific areas of the brain. This data is continuously recorded and used to identify abnormal brain patterns that may lead to seizures.

2. Detection Phase: When the device identifies abnormal brain activity similar to seizure precursors, it enters the detection phase. In this stage, the device evaluates the brain activity and determines whether this activity can potentially lead to a seizure or not.

3. Stimulation Phase: In this phase, after identifying abnormal brain activity, the device sends electrical pulses to the brain. These stimulations are designed to prevent the onset of a seizure or to limit its spread to other areas of the brain. These pulses can act preventively and prevent the occurrence of a seizure or stop the seizure in its early stages. Despite the three-phase structure of monitoring, detection, and stimulation in commercial RNS systems, a unified and formal mechanism for runtime validation and intelligent adaptation of the device's decisions under critical conditions—especially one that systematically exploits multisensory brain and physiological information—has not yet been established; this gap is addressed in the present work through a hybrid TCPN–fuzzy model combined with hierarchical reinforcement learning.

2.2. Colored Timed Petri Nets

Petri net is a graphical language that was produced and introduced by Carl Adam Petri to express and describe processes. One of the most important advantages of Petri nets is the ability to use formal methods for analysis. General Petri nets are overly abstract and cannot be a suitable tool for specialists in some applied fields. Colored Petri nets, which were developed by Jensen and Christensen (2009), come with a functional programming language that can make the use of Petri nets easier and simpler [6]. A colored Petri net is a modeling system that allows for its extensive application not only for displaying system capabilities and modeling it but also for programming and control. Petri nets consist of two main elements: places and transitions. Places, which are represented by circles, indicate the states or resources of the system and can contain tokens that specify the system's status. Transitions, which are represented by rectangles or bars, model events or processes that cause changes in the system's state. Directed arcs between places and transitions determine the flow of tokens. In colored Petri nets, tokens can carry complex data such as colors or values, which provides the possibility for more precise and practical modeling. This structure enables the analysis and simulation of complex systems with high accuracy. To show an example of a Petri net, the Petri net of the interaction of the three main phases in the execution of the RNS device with each other has been used, the structure of which is presented according to Figure 2.

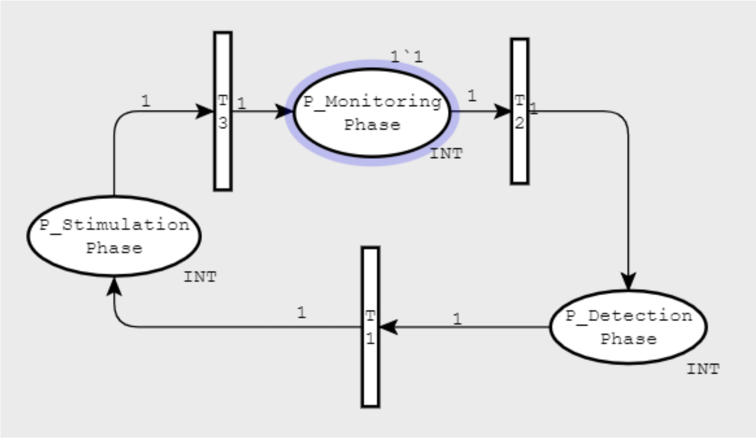


Figure 2 - Petri Net of the Phases of the RNS Device

As illustrated in Figure 2, the proposed Timed Colored Petri Net (TCPN) model comprises several main places that represent the key phases of the device, including places for monitoring, feature extraction and signal analysis, detection, therapeutic decision-making, stimulation, and finally validation of the device's behavior. Important transitions such as "Abnormal Activity Detected," "Seizure Onset," "Therapy Delivered," and "Validation Feedback" govern the state changes between these places and describe the temporal evolution of brain states and control decisions.

2.3. Hierarchical Reinforcement Learning

Hierarchical reinforcement learning (HRL) extends classical RL by decomposing a long-horizon decision problem into a hierarchy of sub-tasks operating at different temporal and semantic scales. Instead of learning a single flat policy over primitive actions, HRL introduces higher-level controllers that select temporally extended options or sub-policies, each of which may execute multiple low-level actions before terminating. This idea is formalized in the options framework of Sutton et al., where an option is defined by an initiation set, an intra-option policy, and a termination condition, and planning is performed in a semi-Markov decision process over these abstract actions [7].

A rich body of work has shown that such hierarchical decompositions can significantly improve sample efficiency, exploration, and transfer in complex domains. Barto and Mahadevan provide a comprehensive survey of HRL methods and argue that temporal abstraction and state abstraction are key mechanisms for scaling RL to real-world tasks [8]. Dietterich's MAXQ framework formalizes HRL as a decomposition of the target Markov decision process into a set of smaller MDPs together with an additive decomposition of the value function; the MAXQ-Q algorithm is proven to converge to a recursively optimal hierarchical policy and has been shown to learn much faster than flat Q-learning on several benchmarks[9]. More recently, HRL ideas have been combined with deep function approximation, where a high-level policy chooses goals or sub-tasks and lower-level deep networks learn to realize them in high-dimensional state and action spaces. For example, Kulkarni et al. propose hierarchical DQN (h-DQN), in which a top-level controller selects intrinsic goals and a low-level controller learns goal-conditioned Q-functions; they demonstrate improved exploration and performance in sparse-reward environments such as Montezuma's Revenge [10].

In line with these developments, our work adopts an HRL architecture tailored to runtime validation of a brain neural stimulation (RNS) device. Building on a timed colored Petri net (TCPN) model of the device workflow and a fuzzy risk-assessment layer, we define a two-level hierarchy: a high-level manager that, at a coarse time scale, selects among "no intervention", "delegate to stimulation worker", and "inhibit stimulation", and a low-level worker that, when activated, generates continuous stimulation patterns based on brain and physiological sensor inputs. This design follows the general HRL principle of separating "what to do" (high-level clinical strategy over risk states) from "how to do it" (fine-grained control of stimulation parameters), while remaining compatible with established HRL theory on temporal abstraction and value-function decomposition [11].

2.4. Fuzzy Logic

Fuzzy logic is a mathematical framework for modeling and reasoning under conditions of uncertainty and ambiguity. Instead of assigning variables to crisp sets, it associates each variable with a membership degree in the continuous interval [0,1], representing the extent to which that variable belongs to a given linguistic category (e.g., *Low*, *Medium*, *High*)[12]. This representation enables more realistic modeling of systems whose input data are imprecise, noisy, or strongly context dependent. In the present problem, the signals obtained from brain sensors have an inherently uncertain nature due to measurement noise, momentary variability, and inter-patient differences. Therefore, these signals are first mapped to fuzzy linguistic values using suitable membership functions and then injected into the timed colored Petri net structure as fuzzy rules. This combination, in addition to providing flexibility for analyzing diverse operating conditions, allows complex multimodal sensor data to be integrated directly into the decision-making process and paves the way for improving the accuracy of validation and the responsiveness of the system to critical conditions at runtime. Similar integrations of fuzzy logic with (colored) Petri nets have already been shown to be effective for medical diagnostic and decision-support systems under uncertainty, which supports the suitability of the proposed modeling approach in this work. Similar ideas of integrating fuzzy logic with hierarchical colored Petri nets for runtime verification of implantable medical devices have been successfully applied in our previous work on pacemaker functionality validation [13].

3. Research Background

In this section, the background of the research topic and similar solutions that address the problem of this research are examined. For a more precise examination, the research background is reviewed in four categories. The categorization of the background into four categories is based on the domain of each and considering the thought path for examining the topic and the proposed solution.

- **First Category - Review Articles in the Research Domain**

Runtime validation of medical devices is an important topic in the field of wearable devices. Due to the expansion and use of these devices in the body internet, examining methods and techniques that can investigate and improve their functional and security challenges is necessary. Therefore, this category of articles has been reviewed to demonstrate the importance of the current research conducted in the field of medical device validation.

Based on a comprehensive review conducted in [14], it has been determined that WBANs face several operational, standardization, and security problems that affect performance and the preservation of user safety and privacy. However, the dependence of future healthcare devices on WBAN for medical and non-medical applications is inevitable. Despite the advantages that WBANs have in remote health monitoring, more studies need to be conducted to optimize performance and address open issues in improving their performance. This study highlights the importance of working on the security and validation of medical devices, which is the main topic of this research article. The overall trend in IWMDs is towards increasing functional complexity, software programmability, and wireless network connectivity. This issue creates an undesirable yet inevitable situation where IWMDs and BANs become increasingly vulnerable to security attacks [3]. In [3], various aspects of the threats that IWMDs face are analyzed, and appropriate solutions for each threat are discussed and examined. Given the vital functions of IWMDs, these issues must be seriously and preventively resolved by manufacturers before market release. The use of validation methods, including the solution used in this research, is one of the preventive solutions.

Ensuring that the data sensed and collected by WBAN sensors is secure and not exposed to unauthorized entities and security threats is of great importance. Therefore, strong security solutions and authentication schemes are needed for the success and large-scale acceptance of WBANs. For this purpose, a set of security solutions and authentication schemes have been proposed by researchers over the past two decades. In [15], first, an extensive review of essential security requirements, security threats, attackers and attack techniques, and current existing solutions with precise classification of security mechanisms in WBANs is presented. In the second stage, a detailed discussion on authentication, design and development of authentication schemes and their classification, models, and verifiers of the presented security protocols is provided. In addition, this article describes applications, open research issues, and provides recommendations for authentication schemes for WBANs.

- **Second Category - Security of Medical Devices and Patient Medical Data**

The articles reviewed in this category examine the security issues of medical devices. Medical devices, due to their wireless connection to the network, create opportunities for penetration for malicious attackers. Given that these devices face limitations in power and memory size, using conventional security solutions such as encryption in them is impossible.

In [2], a medical security monitoring method called MedMon is proposed that monitors all radio frequency wireless communications to medical devices and uses anomaly detection to identify malicious transactions. After identifying a malicious transaction, it performs appropriate response actions, which can range from passive methods (notifying the user) to active methods (collecting packets in such a way that they do not reach the medical device). The advantage of this method is that it performs this without any hardware or software changes in the medical device. The proposed method presented in this research focuses monitoring on the received information rather than on wireless communications and does not focus on wireless communications.

Many remote healthcare patients monitoring protocols that have been seen so far are vulnerable to many attacks, including replay attacks and impersonation attacks, and preserving privacy in them is a dilemma. Therefore, in [16], an improved scheme and a new healthcare monitoring protocol for patients is presented that formally simulates the features of medical devices using first-order logic. The idea of formally examining the behavior of medical devices and validating based on it is also used in the proposed method of this research.

In [17], considering the importance of implementing appropriate security measures, a security solution is proposed to increase the safety of IoMT against cyber-attacks. This security approach increases the execution safety of the medical device. In the proposed method of this research, the safety approach to counter attacks is resolvable and addressable through reinforcement learning.

- **Third Category - Articles Dealing with Testing and Verification of Medical Software**

In this category, articles and research that aim at testing and verifying the software existing in a medical device are examined. In the proposed approach of this research, the software used in the medical device is also validated, so this category has the most similarity in the proposed approach with this research.

In [18], a framework for testing medical device software is presented. Given that most medical devices are continuously connected to the network and cloud environment, and the programs in them change frequently, therefore, to emphasize the quality and reliability of the system after changes, regression testing during their changes is necessary and essential. In this article, a design pattern is proposed to increase the fault detection rate. This proposed framework increases the fault detection rate compared to previous error-based and random priority approaches.

In [19], formal methods are used for modeling and verifying a pacemaker device. The pacemaker is an electronic device that controls and regulates the heart rhythm through sensing and pacing operations. Precise verification and proof of the correctness of this device's software with respect to its specifications leads to reliable software and reduces potential costs. To do this, three stages were performed. The first stage uses formal methods based on timed colored Petri nets (TCPN) for modeling and verifying the interdisciplinary requirements of pacemaker systems. The second stage uses formal methods to accurately present and generate realistic artificial events of electrical heart activities for verifying various pacemaker parameters. Finally, and in the third stage, a systematic formulation for building appropriate and relevant datasets from the processed data of the above models for analysis, optimization, and further application.

In [20], a verification method for WBSNs operating in healthcare systems is presented. In this method, Event-B is used to model the design and requirements of a WBSN, and then its correctness is verified using the underlying tool. In this article, several functional features in WBSNs that are difficult to verify using conventional simulation methods are examined. For example, verifying whether the WBSN sends vital values after detecting a critical value or after completing a time period, regardless of the categorization of vital values, is investigated.

- **Fourth Category - Articles Dealing with Modeling Medical Domains with Colored Petri Nets**

In this section, articles that use types of Petri nets to model the behavior of medical devices for monitoring or examining the correctness of performance are examined. It also includes some previous works by the authors of the article.

Article[13] seeks a method for monitoring the software behavior of a medical device to reduce the risk of failure. This is done by examining the runtime performance and software status of that device. In this article, a method for verifying pacemaker software based on the device's fuzzy performance is presented. The device's functional limitations are identified and presented as fuzzy rules, and then the device is verified based on a hierarchical fuzzy colored Petri net (HFCPN) that is formed according to the software limitations. HFCPN in this article reduces the runtime verification time by 90.61% compared to the runtime verification time in similar previous works.

Article [21], which is an extension of article[13], presents an automated runtime method for continuous verification of implanted pacemaker behavior using an intelligent software agent. The independence and intelligent features of the software agent for controlling the pacemaker software behavior are employed through inference from a knowledge base in which HFCPN is used by the agent as an inference engine. Compared to a flat inference engine, HFCPN can cover concurrent states initiated by fuzzy input values and improve the runtime for finding an appropriate rule by up to 92%. In addition, the intelligent software agent checks the runtime performance accuracy of the pacemaker software in critical and unexpected conditions and, if an unacceptable value is found, changes the software decision.

In [22], considering rules with fuzzy variables, MFCPN (multi-level fuzzy colored Petri nets) is designed for modeling rule-based knowledge to systematically represent rules and infer them in places where appropriate action is necessary. For unforeseen situations, decisions are made using self-learning, and these decisions are corrected using feedback to the device through reinforcement learning. The proposed approach is applied to the pacemaker to effectively examine and observe the impact of this approach. The effectiveness of the proposed model was evaluated through several scenarios on the case study and its dataset. Comparison between the proposed method and two real datasets showed that the proposed method has 79.3% accuracy.

In [4], a colored Petri nets (CPN) model of a surface electrocardiography device is presented to help manufacturers increase their confidence during device development and certification processes. This work is important because the electrocardiography device is classified as a Class II device. There are standard requirements to increase confidence in the execution of this device. Formal methods such as colored Petri nets can increase this confidence.

In[23], high-level colored Petri nets are used for modeling preemptive scheduling in real-time systems that require concurrent access in true parallelism to shared resources. This approach is used for modeling a preemptive multi-core concurrent program to examine all possible execution paths, service access conflicts, and preemptive scheduling. Timed automata and simple timed Petri nets do not allow direct representation of these features.

In article[24], a supervisory system for validating ECG devices is presented. Medical information receiving devices (e.g., electrocardiography-ECG) that are used to diagnose and monitor individuals suffering from various diseases such as cardiovascular diseases are vital and important devices that must comply with regulatory requirements before market release to prevent unauthorized access or operation, and their functional correctness must be precisely examined, and preventive methods used to check their correct operation.

- **Fifth Category - Reinforcement Learning and Hierarchical Reinforcement Learning in Healthcare and Neuromodulation**

In recent years, reinforcement learning (RL) has been increasingly investigated as a framework for clinical decision support and dynamic treatment optimization in high-risk medical settings. Several comprehensive reviews conclude that RL can learn individualized treatment policies from retrospective clinical data for tasks such as sepsis management, hemodynamic control, and ventilator weaning, but also highlight outstanding challenges in safety, interpretability, and offline evaluation of learned policies [25-27]. Importantly, most of these RL-based systems operate at the level of high-level clinical decisions (e.g., drug dosing, ventilator settings) and do not explicitly model the internal dynamics of implantable or wearable medical devices.

In a more device-centric line of work, RL has been used to design adaptive closed-loop neurostimulation controllers. Early studies demonstrated that RL can learn stimulation policies that reduce seizure frequency and duration in computational models of epilepsy by modulating stimulation parameters as a function of observed neural activity [28, 29]. More recent contributions propose RL-based controllers for deep-brain stimulation and epilepsy therapy that adjust stimulation amplitude or timing based on neural biomarkers and are trained either in silico or with offline clinical data, with the goal of improving efficacy and energy efficiency compared to conventional open-loop stimulation [30, 31]. These approaches confirm the feasibility of RL-driven neuromodulation, but they neither incorporate formal models such as timed or colored Petri nets nor exploit body-area sensor networks within a unified runtime-validation framework.

Hierarchical variants of RL (HRL) have also been explored to cope with complex medical decision spaces. For example, hierarchical RL has been applied to automatic disease diagnosis, where a high-level policy selects which symptom or test to query and low-level policies determine concrete diagnostic actions, leading to improved diagnostic accuracy and symptom recall over flat RL baselines [32]. Other work proposes hierarchical RL with a decomposed action space and recurrent state representation to allocate scarce medical resources during the COVID-19 pandemic under imperfect and delayed information [33]. More recently, dialogue-based disease diagnosis systems have combined hierarchical RL with multi-expert feedback to structure diagnostic conversations and integrate external probabilistic symptom information [34]. However, these HRL approaches primarily target clinical workflows, epidemic interventions, or dialogue systems and do not address the runtime validation and adaptive control of implantable or wearable medical devices modeled by TCPN and fuzzy rules.

Considering the articles reviewed in this section, the research gap and the innovation of the proposed solution in this research are shown in the axes of Figure 3. In two of the four axes of this figure, the proposed method of this article has innovation, which is marked in red in the Figure. (1) First axis: runtime. Medical devices are examined and evaluated by various testing and verification methods before use in the human body. These methods are static and not during the actual runtime of the device in the human body. Some research conducted so far has emphasized these methods and performs validation statically. While in the proposed approach, it provides assurance and validation capabilities at runtime and dynamically examines the medical device's performance. 2) Second axis: learning capability. The proposed approach employs a hierarchical reinforcement learning (HRL) agent, in which a high-level manager policy selects abstract options (e.g., monitoring, stimulation, inhibition) and low-level controllers execute these options over several time steps and update the device behavior accordingly. In contrast to other approaches and previous research, which rely on a fixed knowledge base or pre-defined rules embedded in the device – rules that remain unchanged regardless of the agent's context and operating conditions – the proposed HRL-based agent can continuously learn from experience and adapt its decision-making structure. Thanks to the hierarchical reinforcement learning mechanism embedded in the intelligent agent, the knowledge base and decision rules can be refined and corrected in response to new, rare, and unforeseen situations observed at runtime. 3) Third axis: modeling. The proposed approach uses TCPN modeling, which, in addition to being timed and considering the impact of time in it, covers the non-stationary conditions. This modeling is colored. In similar methods and approaches, modeling has been in the form of HFCPN or FCPN. 4) Fourth axis: validation. The proposed approach performs validation dynamically in the fourth dimension, which is similar to a limited number of previous approaches that also perform dynamic validation. To the best of our knowledge, this combination of HFCPN-style modelling with hierarchical RL for online safety monitoring and decision correction in an IWMD/BAN setting has not been addressed in the existing literature.

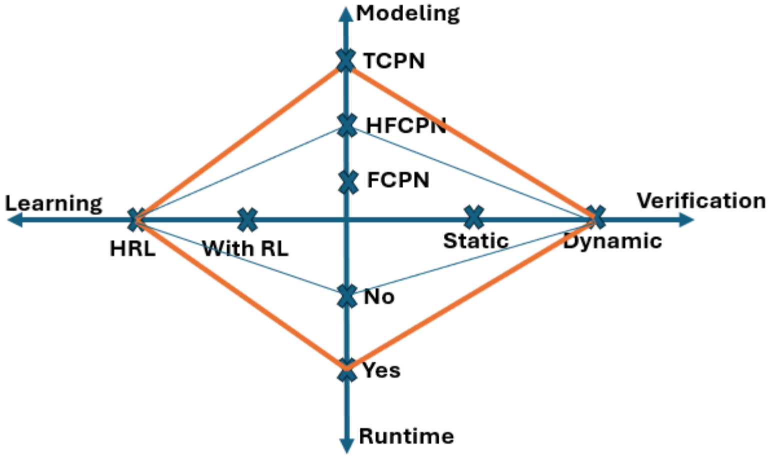


Figure 3 - The Position of This Research's Innovation in Comparison with Other Research

4. Proposed Model

The structure and position of the proposed method in this research can be observed in Figure 4. As shown in the figure, the implantable internet-of-the-body (IoB) device, installed in the patient's body, uses relevant sensors to examine the patient's physical condition and, based on the type of device performance, performs the necessary medical care or treatment. The data received from the sensors, along with the output produced by the body medical device, enters the proposed validation section. In this section, first, the received parameters are entered into the input unit, where initial preprocessing is performed on the data. The input data are examined, standardized, and if necessary, fuzzified. These values are then transferred to the inference unit in the second stage, where the inference unit evaluates the expected system output using the knowledge base rules that are modeled based on HTCPN. Then, in the decision-making unit, a decision is made regarding the validity of the output produced by the body medical device and the correctness of the output obtained from the validation system. By applying the output to the patient, the expected feedback is received and entered into the learning unit. In the learning unit, based on feedback, rule corrections are made, and if necessary, the knowledge base is updated. For evaluating the proposed method, 10 reference scenarios together with additional noise-augmented scenarios are used to train the hierarchical agent, as will be described in the Evaluation section.

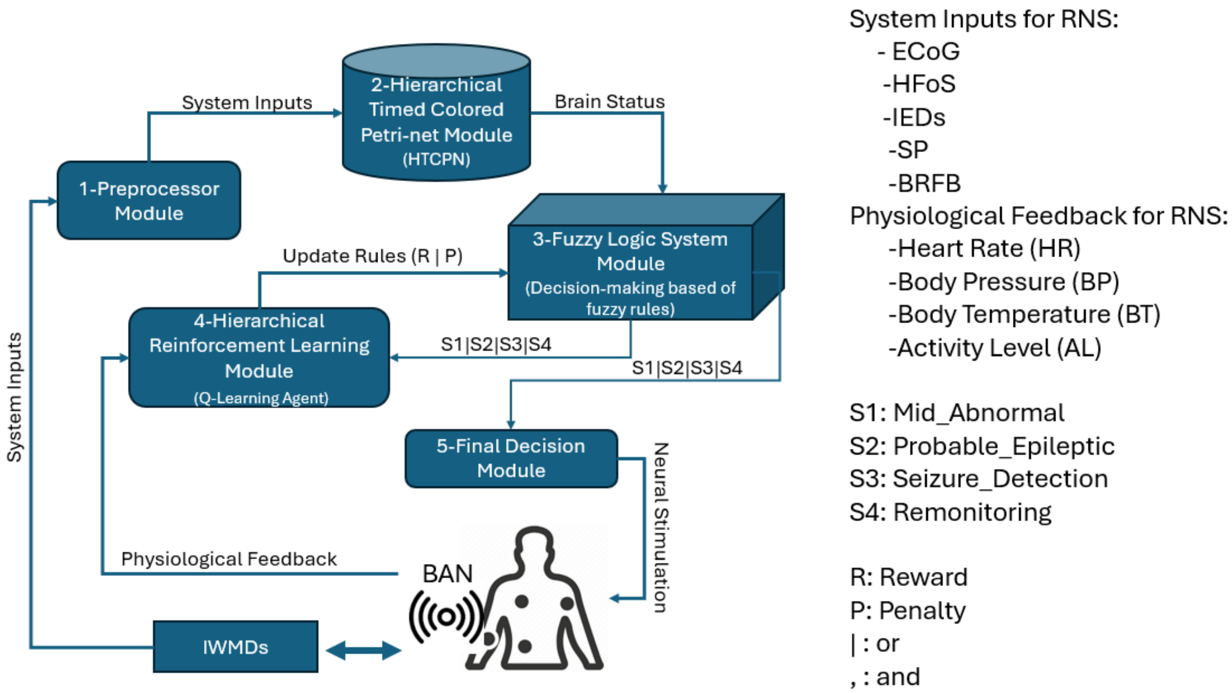


Figure 4 - Structure of the Proposed Method

The proposed model is repeated in a loop at two main times T1 and T2. In each of these times, several tasks are performed simultaneously. At time T1, stages 1, 2, and 4 are performed, and at time T2, feedback from the environment is received. The time interval between T1 and T2 is such that it allows applying the output to the body and receiving feedback from the body. Again, at time T3, which is a repetition of time T1, stages 1, 2, and 4 are performed in such a way that the time interval between T3 and T2 is sufficient for updating the knowledge base and correcting the model. Therefore, the time difference between T2 and T1 and the time difference between T3 and T2 are as follows:

T2 - T1 > Applying output to the body and receiving feedback
T3 - T2 > Updating the knowledge base and correcting the model

The proposed method is performed in two stages. The first stage is model preparation, and the second stage is model execution. The model preparation stage, which is performed only once at the beginning of the design and creation of the model, is done offline using existing datasets to design the model. In the second stage, the model designed in the previous stage is executed and performs the validation of the medical device. In this stage, real-time data obtained from the patient's body is used.

For a clearer expression of the above pseudocode, a formal expression is used, which will be addressed below. The formal expression of the proposed structure in the input space is as follows:

$I=\{ I_{Brain} ,\ I_{Physio} \}$

where:

$I_{Brain}=\{$ ECoG,HFOs,IEDs,SP,BRFB $\}$

$I_{Physio}=\{$ HR,BP,BT,Activity $\}$

Each sensor has fuzzy values (Low, Medium, High) that are obtained using membership functions as in Equation 1:

$$\mu S_i(x) : \mathbb{R} \rightarrow [0.1]\ .\ \ \forall S_i \in I.\ x \in Range(S_i) \qquad (1)$$

where $\mu S_i(x)$ is the membership function for sensor $S_i$and x is the sensor value.

The formal expression of the timed Petri net is as a 7-tuple in the form of Equation 2.

$$\text{TCPN}=(P.T.A.\textstyle\sum.C.G.E) \qquad (2)$$

where P defines the set of places, T defines the set of transitions, and A defines the set of arcs that indicate the connections between places and transitions.

P={Monitoring, Detection, Stimulation}

T={Normal_Activity_Detection,Probable_Epileptic_Activity_Detection, Seizure_onset_Detection}

$A \subseteq (P \times T) \cup (T \times P)$

Σ is the set of colors representing the fuzzy values of the sensors, based on which C is the function assigning colors to places, and G specifies the enabling conditions of transitions based on fuzzy rules, and E is the arc weight function that controls the data flow between places.

$C{:}P\rightarrow\sum.$

$G{:}T\rightarrow Expre$

$E{:}A\rightarrow Expre$

The formal expression of the fuzzy system including the fuzzy rule base is defined as Equation 3.

$$R=\{r_1,r_2,..,r_n\}, \qquad n=16 \qquad (3)$$

Each rule r_i‘ is as follows:

$r_{i:}$ IF (S_1 is M_1)∧… ∧ (S_{ki} is M_k) THEN Output is O_i

where

$M_i \in \{Low.Medium.High\}$

$$O_i \in \begin{cases} normal. \\ Probable\_Epileptic. \\ Seizure\_Detection \end{cases}$$

The formal expression of reinforcement learning based on the reward received from the system is expressed as Equation 4.

$$Q(s_t.a_t) \leftarrow Q(s_t.a_t) + \alpha[R(s_t.a_t) + \gamma \max_{a'} Q(s_{t+1}.a') - Q(s_t.a_t)] \qquad (4)$$

where in this equation α is the learning rate, γ is the discount factor for future rewards, s_t, a_t is the state and action at time t, $s_{t+1}$is the next state after performing the action. The reward used in the proposed method will be described later. The finite state machine (FSM) of the system execution is depicted as Figure 5.

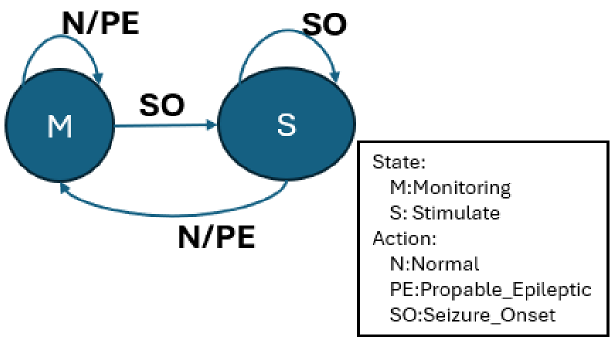


Figure 5 - Finite State Machine of the System

As observed in Figure 5, there are two stages: monitoring and stimulation operation. In the monitoring state, if a seizure is detected, it enters the stimulation operation state and remains in this state as long as the seizure still exists. If in the seizure state, probable seizure or normality is detected, it returns to the monitoring state to examine the status of brain parameters.

4.2. Fuzzy System Structure

In the proposed method, for validating the behavior of the RNS medical device, a colored Petri net is used. This Petri net receives the necessary data from the BAN based on five brain sensors. These five sensors are described in Table 1, and due to the fuzzy nature of the values, the inputs of these five sensors are fuzzified. For each sensor, three areas with trapezoidal or triangular areas are considered, and the values of each area can be observed in Table 2 and the output of the fuzzy system shows in Table 3. For example, the fuzzy diagram of two sensors ECoG and SP can be seen in Figure 6. The fuzzy system output is observed in the table. Based on the fuzzy division of these 5 sensors, the number of rules is 243 (3^5), many of which will be impossible due to the nature of the sensor values, and this number is reduced to 16 rules. The complete list of these rules is in Appendix 1. Three sample rules are as follows.

Table 1 - Brain Sensors

Row	Sensor Name	Description
1	ECoG	Cortical electrical potentials that are used to record high-resolution brain electrical activity showing neuronal changes in neural activity patterns.
2	HFOs	Represents high-frequency oscillations. Fast and high-frequency waves that are often observed in epileptic areas and may indicate epileptic activity. An increase in HFO can be a sign of the area where the seizure starts.
3	IEDs	Represents pre-seizure epileptic potentials. Short-term abnormal electrical activities that occur between seizures and may include spikes and sharp waves. The presence of IEDs indicates epileptic activity and is more observed in patients with severe epilepsy.
4	SP	Sudden changes in amplitude and frequency of brain signals that indicate seizure onset.
5	BRFB	Represents changes in brain rhythms. Brain waves (delta, theta, alpha, beta, and gamma) are divided into different frequency bands, and changes in them can indicate normal or abnormal brain activity. Changes in these bands may be a sign of abnormal brain activity or seizure.

Table 2 - Fuzzy Values of Brain Data Sensors

Row	Sensor Name	Value Range	Fuzzy Area	Fuzzy Area Range
1	ECoG	[0 100]	Low	[0 0 25 50]
			Medium	[25 50 75]
			High	[50 75 100 100]
2	HFOs	[0 300]	Low	[0 0 50 100]
			Medium	[50 100 200]
			High	[200 250 300 300]
3	IEDs	[0 20]	Low	[0 0 3 6]
			Medium	[3 6 10]

Row	Sensor Name	Value Range	Fuzzy Area	Fuzzy Area Range
			High	[10 15 20 20]
4	SP	[0 1]	Low	[0 0 0.05 0.1]
			Medium	[0.05 0.1 0.15]
			High	[0.1 0.15 1]
5	BRFB	[0 50]	Low	[0 0 10 20]
			Medium	[10 20 30]
			High	[30 40 50 50]

Table 3 - Fuzzy Values of Fuzzy System Output

Value Range	Fuzzy Area	Fuzzy Area Range
[0 1]	Normal	[0 0 0.3 0.55]
	Probable_epileptic	[0.3 0.55 0.75]
	Seizure_Onset	[0.55 0.75 1 1]

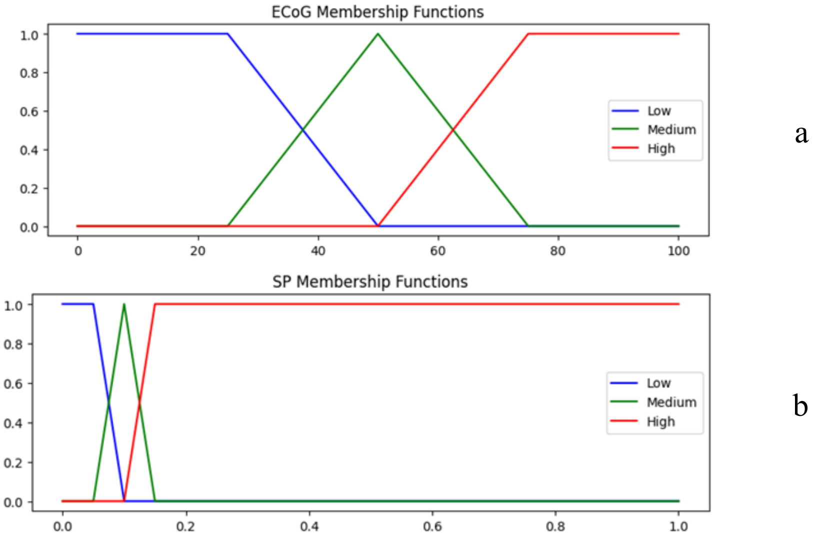


Figure 6 - Fuzzy Diagram of Two Sensors ECoG and SP

1) "If SP is Low then Output is Normal";

2) "If SP is High and IEDs is High and HFOs is Medium then Output is Mid_Abnormal_Activity_Detection";

3) "If HFOs is Medium and BRFB is High and ECoG is High then Output is Seizure_onset_Detection";

In the first rule, if the SP sensor value is low, the other sensor values are irrelevant, and due to the invalidity of the sensor input, monitoring should be redone. In the second rule, if the HFOs value is Medium and BRFB and ECoG values are High, the output is mild abnormal activity detection. In the third rule, if the HFOs, BRFB, and ECoG sensor values are High, the output is definite seizure detection. The code for creating this fuzzy system is available in Appendix 2. These rules are used for designing the colored Petri net, which is described next.

4.3. Hierarchical Timed Colored Petri Net Structure

For the execution of the proposed model, the RNS device behavior is specified as a hierarchical timed colored Petri net (TCPN). As illustrated earlier in Figure 2, the RNS device operates in three main modes: Monitoring, Detection, and Stimulation. In the HTCPN, these three modes appear as three places at level zero, and the flow of control between them is shown in Figure 7. Figure 7 depicts level zero of the HTCPN, while Figure 8 refines the structure at level one. For clarity, the reinforcement learning part of the model is omitted from Figure 7.

Each brain sensor considered in the previous sections is represented by an input place in the colored Petri net. These five input places feed three key transitions: Mid_ Abnormal_ Activity_ Detection, Probable_Epileptic_Activity_Detection, and Seizure_ onset_ Detection. The firing conditions of these transitions are obtained from the fuzzy rules designed in the fuzzy-logic subsystem, and the corresponding thresholds for the three epileptic risk levels are summarized in Table 4. If one of the first two transitions fires, the device returns to or remains in the Monitoring place and continues to observe the patient. In this case, after a time delay of 2000 time units, the sensor values are sampled again and the conditions for abnormal activity are re-evaluated.

When the third transition Seizure_onset_Detection fires, the TCPN moves to the Stimulation place, which represents a critical condition. In this mode, the RNS device delivers targeted electrical pulses to specific regions of the brain to suppress or terminate the detected abnormal activity. This intervention is crucial to prevent the spread or escalation of the epileptic episode and constitutes a key component of the treatment process. After stimulation has been applied, the system either returns to the monitoring state after a specified delay or remains in stimulation depending on the observed brain response. The activation signal from the TCPN is passed to the neural stimulation control module, which drives the implanted electrodes in the target region and automatically applies inhibitory pulses.

Because these interventions directly affect the patient's physiological state, it is important to monitor how the RNS device influences the body and the brain signals over time. For this purpose, a set of reinforcement learning (RL) parameters is introduced that track the physiological changes induced by stimulation; these parameters are aligned with the effects of the RNS on the human body and will be detailed in the next subsection. When the TCPN enters the Stimulation place, control is handed over to a hierarchical reinforcement learning agent. At that point, a high-level manager decides whether to continue monitoring, to trigger the stimulation worker, or to inhibit stimulation, while the worker executes the chosen stimulation pattern. The resulting effect of the stimulation is propagated back to the TCPN through the action effect component of the state-update function, and the full structure of this hierarchical controller is described in Section 4.4.

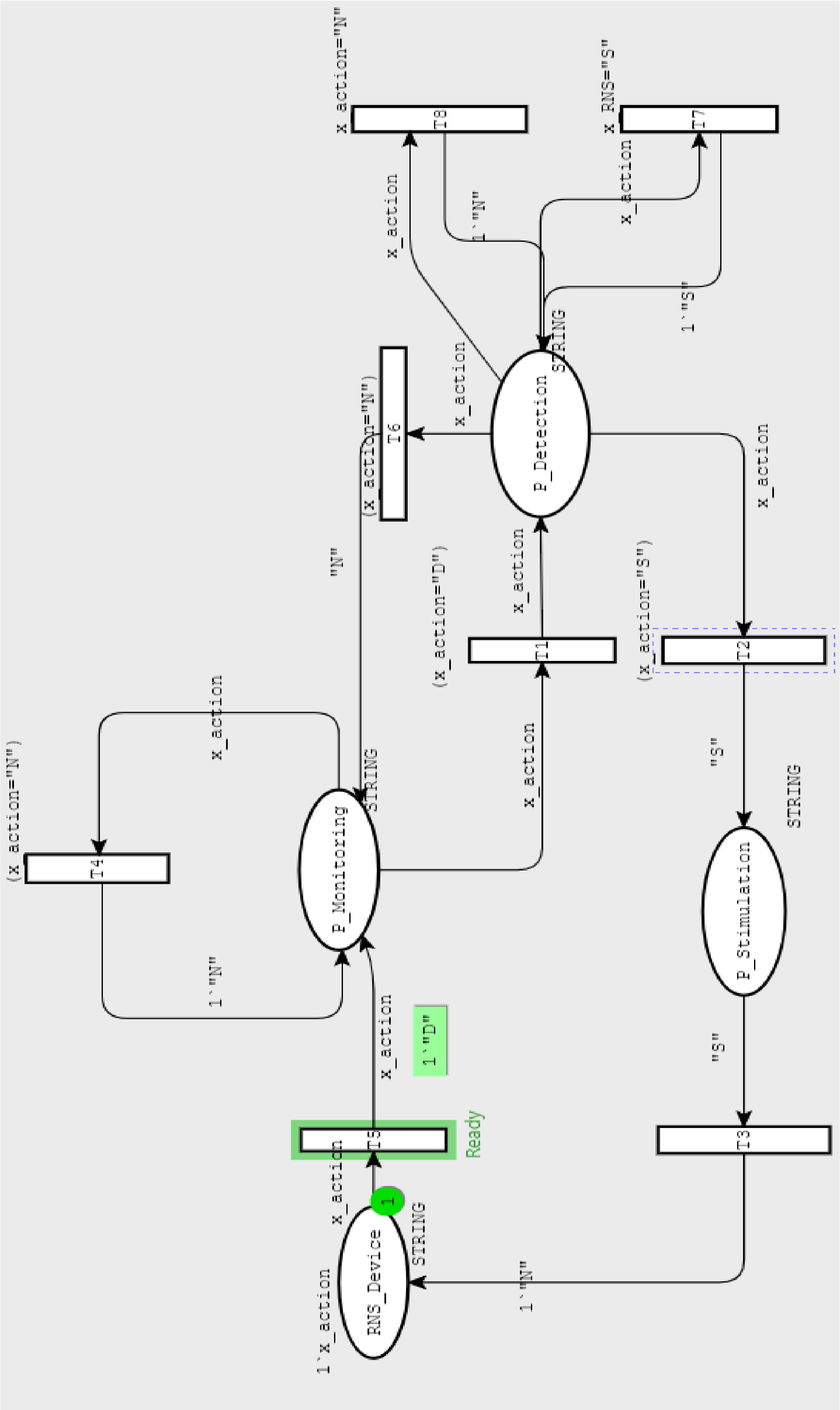


Figure 7 - Level Zero Colored Timed Petri Net

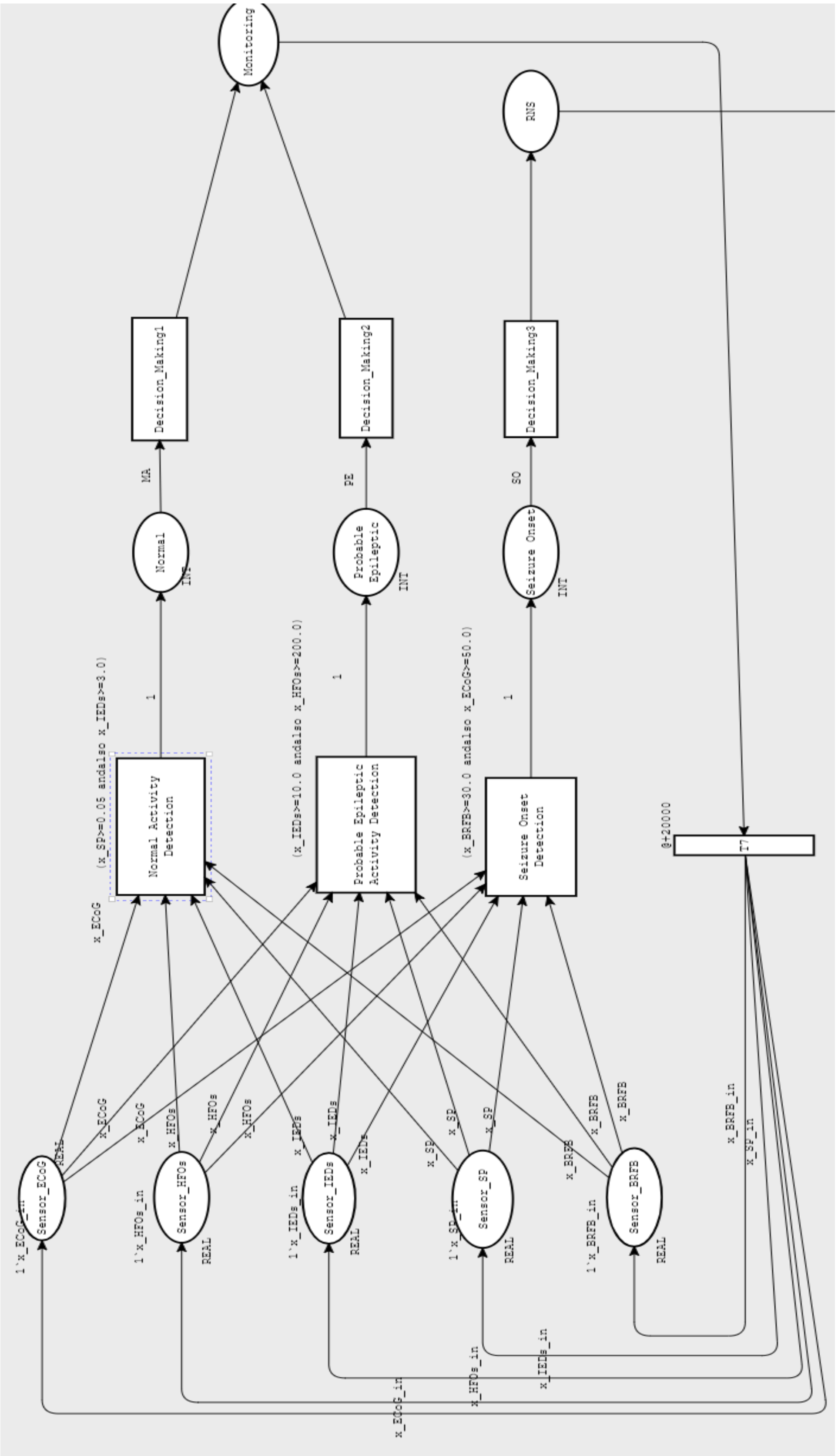


Figure 8 - Level One Colored Timed Petri Net

Table 4 - Conditions Considered for Each Transition

Row	Transition Title	Execution Condition
1	Mid_Abnormal_Activity_Detection (Mild Abnormal Activity Detection)	(x_SP>=0.05 andalso x_IEDs>=3.0)
2	Probable_Epileptic_Activity_Detection (Probable Epileptic Activity Detection)	(x_IEDs>=10.0 andalso x_HFOs>=200.0)
3	Seizure_onset_Detection (Definite Seizure Detection)	(x_BRFB>=30.0 andalso x_ECoG>=50.0)

4.4. Hierarchical Reinforcement Learning Parameters and Reward Design

The reinforcement learning component uses physiological variables as its main inputs, so that the learning agent can adjust its decisions based on the actual impact of stimulation on the patient's body. The parameters considered in this article are: (1) heart rate, (2) blood pressure, and (3) body temperature. The normal and abnormal ranges for each of these variables are listed in Table 5. Since heart rate strongly depends on the patient's activity level, an additional activity sensor is included. As a result, the RL module observes four physiological signals: heart rate, activity level, body temperature, and blood pressure.

Table 5 - Range of Reward and Penalty for Body Physiological Parameters for Reinforcement Learning

Learning Parameter (Unit)	Normal Range	Abnormal Range (Penalty)
Heart Rate (bpm)	Rest (R): 60-100	Very Low (Bradycardia): Less than 50
	Moderate Activity (M): 100-140	Very High (Tachycardia): Greater than 100 at rest
	High Activity (H): 140-180	Abnormal Response After RNS Stimulation: Sudden increase of 20-30 units from baseline
	Deep Sleep (S): 50-65	
Body Temperature (°C)	Normal (N): 36.1 to 37.5	High (Severe Fever): Above 38.5
	Slightly High (H): 37.5 to 38.5	
Blood Pressure (mmHg)	Normal (N): 90-120	High Blood Pressure: Above 140
	Borderline (Prehypertension) (H): 120-140	Low Blood Pressure: Less than 90

From a hierarchical reinforcement learning perspective, the proposed runtime validator consists of two coupled control levels: a high-level manager and a low-level worker. The manager operates on a slower time scale and observes both the abstract risk state produced by the fuzzy–TCPN layer and the current physiological context. Based on this information, it selects one of a small set of abstract options: continue monitoring, trigger stimulation, or inhibit stimulation. The worker, which is embedded in the device's TCPN/fuzzy control layer, is responsible for executing the selected option over several steps by generating concrete stimulation commands (for example, amplitude, duration, and pattern of stimulation) and feeding their effect back into the TCPN model. In this arrangement, the manager learns when and under which conditions stimulation should be invoked or suppressed, while the worker focuses on the detailed execution of the stimulation pattern corresponding to the current option. This hierarchical decomposition separates high-level safety decisions from low-level control, improves the interpretability of the learned policy, and allows the agent to adapt to new physiological patterns while remaining consistent with the formal TCPN specification.

In the hierarchical reinforcement learning setting used in this work, the manager's state vector consists of the current fuzzy risk level (S1–S3) aligned with the three TCPN outputs(mild abnormal, probable epileptic activity, and seizure onset), concatenated with the four normalized physiological variables described above. Its discrete action space includes three abstract actions: (i) continue monitoring, (ii) delegate control to the stimulation worker, and (iii) explicitly inhibit stimulation.

The Petri net structure dedicated to the RL component is shown in Figure 9 as a continuation of the network presented in Figure 8. Within this RL-oriented subnet, a reward $R$and a penalty $P$are defined in terms of the physiological signals and activity level, and are formally expressed in Equations (5) and (6):

$$R = \begin{cases} (x_{AC} = R \ \wedge \ x_{HR} > 60.0 \wedge \ \ x_{HR} \leq 100.0) \vee \\ (x_{AC} = N \ \wedge \ x_{HR} \geq 100.0 \wedge \ x_{HR} \leq 140.0) \vee \\ (x_{AC} = H \ \wedge \ x_{HR} \geq 140.0 \wedge \ x_{HR} \leq 180.0) \vee \\ (x_{tem} \geq 36.1 \ \wedge x_{tem} < 37.5) \vee \\ (x_{BP} \geq 90.0 \ \wedge \ x_{BP} \leq 120.0) \end{cases} \quad (5)$$

$$P = \begin{cases} (x_{HR} < 50.0 \ \wedge \ x_{AC} = R \ \wedge \ \ x_{HR} > 100.0) \vee \\ \qquad\qquad\qquad (x_{tem} > 37.5) \vee \\ \qquad\qquad\qquad (x_{BP} > 120.0) \end{cases} \tag{6}$$

Intuitively, the agent receives a positive reward when the physiological parameters remain within safe ranges consistent with the patient's current activity level and incur a penalty when the values drift into unsafe regions (e.g., tachycardia, hypertension, or elevated body temperature).

Equation (4) in the previous subsection describes the Q-learning update used for the flat baseline agent. In contrast, the hierarchical manager in the proposed model is trained with a Monte Carlo policy-gradient (REINFORCE) algorithm that uses the same reward signal defined by Equations (5) and (6).

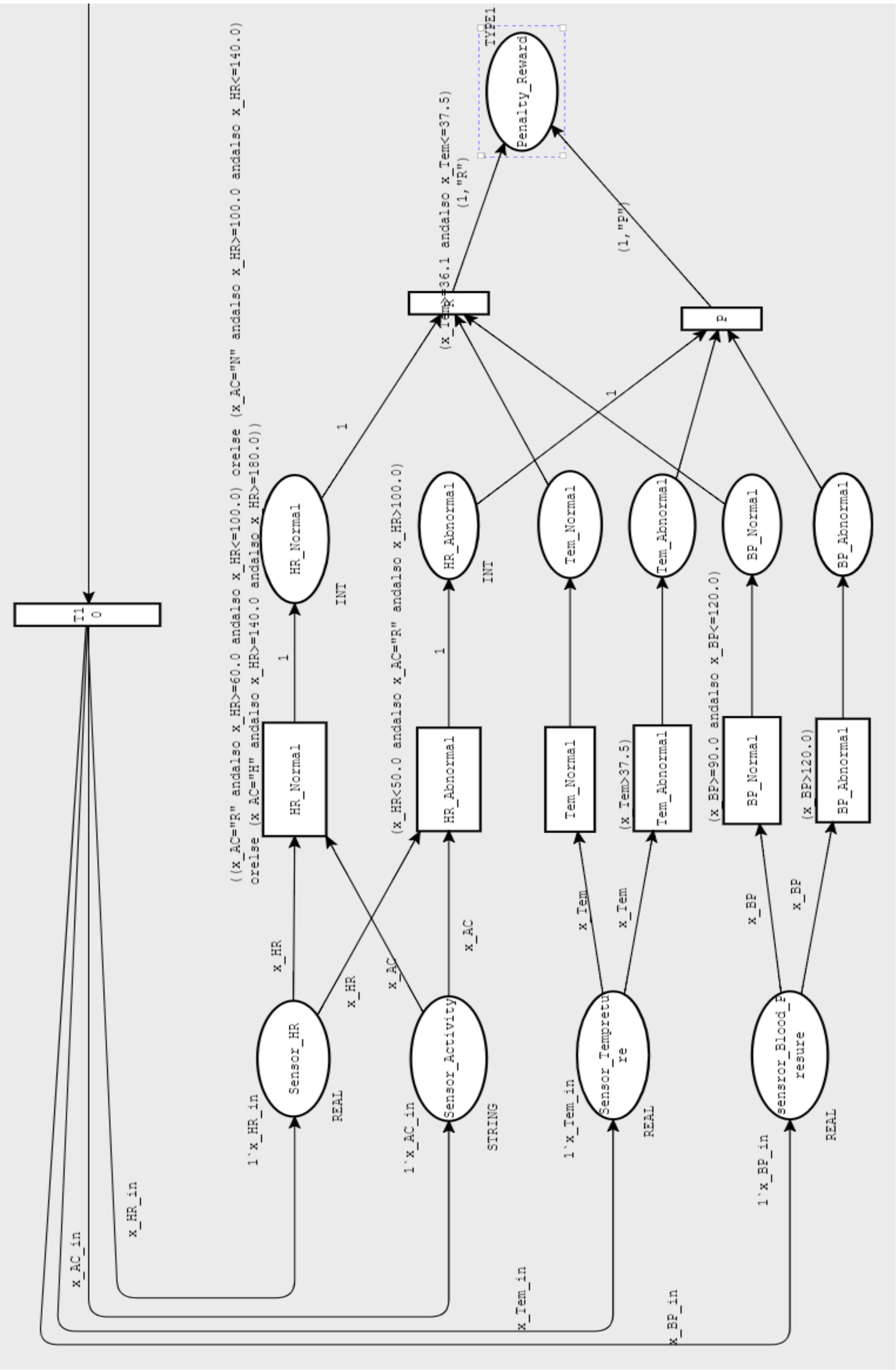


Figure 9 - Level One Colored Timed Petri Net for the Reinforcement Learning Section

5. Scenario Execution

In this section, the behavior of the proposed model is examined using clinically motivated scenarios. First, a set of ten base scenarios was constructed and reviewed with the help of an epilepsy specialist. Each scenario corresponds to a specific combination of brain-sensor and physiological-sensor values together with the expert-desired device response (monitoring or stimulation). These base scenarios cover clear seizure episodes, borderline and ambiguous cases, and normal or quasi-normal conditions, and they form the reference set against which the behavior of the model is evaluated.

Section 5.1 describes three representative base scenarios in detail to illustrate how the fuzzy–HTCPN model and the hierarchical reinforcement learning controller interact. Section 5.2 then explains how additional data were generated from these base scenarios to obtain a richer dataset for training and evaluation of the hierarchical RL agent

5.1 Base Expert-Validated Scenarios

The ten base scenarios were defined using typical and extreme patterns of brain and physiological signals that may occur in patients with epilepsy and were subsequently validated by an epilepsy specialist. Among these ten cases, three representative scenarios were selected to be described in detail here, as they illustrate normal, borderline, and clearly critical situations from the viewpoint of both the expert and the proposed model

Scenario One - Examination of Overt Seizure Status - Initial Detection Correct Status

In this scenario, the patient is in a critical condition, and the system must detect the seizure and apply the necessary stimulation. The initial input of brain sensors is the following values for each sensor.

ECoG=95, HFOs=285, IEDs=18, SP=0.18, BRFB=44

After receiving these values, the fuzzy system output reaches Seizure_onset Detection, which indicates the patient's critical condition and seizure, and the system must apply the necessary stimulation, so the system state enters Stimulation. In this state and after applying the stimulation, the body sensor values for reinforcement learning (activity, heart rate, body temperature, blood pressure) are received. Due to the attack detection and critical conditions, the body enters a stress state, and therefore the body sensor values for heart rate and blood pressure are higher than the normal and resting time limit, and the body temperature also experiences a partial increase. In this situation, the system receives a negative reward (penalty) and therefore remains in the same Stimulation state (see Figure 5). With the application of the necessary stimulation, these values gradually decrease, and in the second reception of body parameters, it still receives a negative reward (penalty) and the system remains in the same state again. With the re-application of stimulation, the body enters a normal state and confirms the effectiveness of the stimulation. This change can take from a few seconds to a few minutes. Receiving a positive reward indicates the change of body parameters to a normal state, and the system returns to the monitoring state (M) (see Figure 5). The received values of body physiological sensors are according to Table 6.

Table 6 - Values Received from Body Physiological Sensors in Scenario One

HR (bpm)	Temp (°C)	BP (mmHg systolic)	Activity
130	38.5	150	N
110	38.2	135	N
90	37.8	120	N

Scenario Two - Initial Incorrect Detection, Correction by Learning

In this scenario, the initial input detects the body's condition as normal or quasi-normal, and no brain stimulation is applied, but after receiving the body's physiological data, it becomes clear that the patient's condition is critical and stimulation should be performed. This scenario shows how reinforcement learning can correct the error in the pre-detection stage. The initial values received from brain sensors are as follows:

ECoG=40, HFOs=80, IEDs=5, SP=0.106, BRFB=18

Based on the initial received data, the system estimates the probability of a seizure as low and does not apply stimulation. Now, the values of the learning sensors are received, which are in accordance with Table 7. Upon receiving these values from the body sensors, the model recognizes the patient's critical condition, and despite the initial detection, neural stimulation is initiated.

Table 7 - Values Received from Body Physiological Sensors in Scenario Two

HR (bpm)	Temp (°C)	BP (mmHg systolic)	Activity
140	38.5	150	N
101	37.6	121	H
90	37.5	120	N

Scenario Three - Ambiguous Initial Detection and On the Threshold

In this scenario, the initial detection indicates a probable seizure and is on the threshold, but after subsequent reading of the body physiological sensors for learning, it becomes clear that the body is in a critical condition and requires stimulation accordance with Table 8. This scenario demonstrates the power of learning to correct the initial detection.

ECoG=20.0, HFOs=100, IEDs=5, SP=0.13, BRFB=10.0

Table 8 - Values Received from Body Physiological Sensors in Scenario Three

HR (bpm)	Temp (°C)	BP (mmHg systolic)	Activity
140	38.5	150.0	N
100	38.0	125	N
90	37.6	110	N

The system states are initially detected as Probable_epileptic or probable seizure, and in the end, Stimulation (definite seizure) is obtained in the simulation performed.

5.2 **Synthetic Data Generation**

To enable the hierarchical reinforcement learning agent to generalize beyond the ten expert-validated base scenarios, additional training data were generated through controlled perturbation of the sensor values. For each base scenario, multiple synthetic episodes were created by slightly modifying the brain-sensor and physiological-sensor readings within clinically acceptable ranges, ensuring that the essential clinical meaning of each scenario was preserved.

These perturbations were applied independently to the ECoG, HFOs, IEDs, SP, and BRFB values, as well as to the four physiological parameters (heart rate, activity level, body temperature, and blood pressure). Small random deviations were drawn from probability distributions centered on the expert-approved values (for example, normal or uniform distributions with narrow variance), so that the resulting signals reflect natural variability rather than unrealistic noise.

This process produced a richer set of training trajectories that expose the agent to plausible variations around each base scenario without requiring additional expert annotation. In total, more than 10 synthetic episodes were generated (where 1530 corresponds to the number of repetitions and perturbation combinations applied per base scenario). These expanded datasets were then used to train and evaluate the hierarchical RL agent under a wider range of conditions that closely resemble the variability expected in real-world patient monitoring.

6. Evaluation

To evaluate the performance of the proposed model, a set of quantitative metrics was used to assess its ability to correctly detect seizure and non-seizure conditions, correct initial errors, and ultimately enhance patient safety. For this purpose, ten different scenarios were prepared, and the corresponding brain and physiological sensor values are provided in Appendix 2. The simulation experiments were implemented in Python using the Google Colab environment, and the source code is available at: https://github.com/negarmajma/RNS-HRL.git

For each scenario, the outcome was labelled using the standard confusion-matrix categories:

- True Positive (TP): a real seizure occurs, and the model correctly detects it (stimulation is triggered).
- False Positive (FP): no real seizure occurs, but the model incorrectly detects a seizure and triggers stimulation.
- True Negative (TN): no seizure occurs and the model correctly identifies the condition (no stimulation).
- False Negative (FN): a real seizure occurs but the model fails to detect it (no stimulation).

Based on these counts, the usual performance measures were computed according to Equations (7)–(12)

$$TPR = \frac{TP}{TP+FN} \qquad (7)$$

$$FPR = \frac{FP}{FP+TN} \qquad (8)$$

$$FNR = \frac{FN}{TP+FN} \qquad (9)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (10)$$

$$Precision = \frac{TP}{TP+FP} \qquad (11)$$

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Reclall} \qquad (12)$$

Since the proposed system is adaptive and modifies its behavior based on body feedback in the form of rewards and penalties, high accuracy alone does not necessarily indicate good performance; in some cases, it may simply mean that the model is not changing its initial decisions. For this reason, additional metrics were introduced to explicitly capture the system's ability to correct initial states and to characterize its intervention behavior. In this context:

- the **Correction Rate (CR)** measures the proportion of scenarios in which the final decision differs from the initial decision (a measure of flexibility and willingness to change),
- the **False Correction Rate (FCR)** measures the proportion of scenarios in which a change was made but resulted in an incorrect final decision (i.e., changes that were harmful rather than beneficial),
- the **Correction Coverage (CC)** measures the proportion of initially incorrect decisions that were successfully corrected to match the expert's label,

- and the **Intervention Rate (IR)** indicates the proportion of scenarios that resulted in stimulation.

Formally, letting Ndenote the total number of scenarios, $N_{\mathrm{chg}}$the number of scenarios in which the decision changes after learning, $N_{\mathrm{fc}}$the number of false corrections (initially correct, finally incorrect), and $N_{\mathrm{ic}}$the number of initially incorrect decisions, these quantities are defined as:

$$CR = \frac{TP}{TP+FP} \quad (13)$$

$$FCR = \frac{FP}{TP+FP} \quad (14)$$

$$CC = \frac{TP}{TP+FN} \quad (15)$$

$$IR = \frac{TP+FP}{TP+FP+TN+FN} \quad (16)$$

In our setting, CC coincides with the true positive rate (TPR), because in the initial configuration all seizure cases were misclassified and thus belong to the set of initially incorrect decisions. The evaluation compares two configurations of the system on the same ten scenarios:
(1) the initial configuration before learning, and
(2) the configuration after learning, where the model parameters have been updated based on the reward/penalty feedback from the body and expert-labelled scenarios

Results before learning

In the initial configuration, the results for the ten scenarios are
TP=0, FP=0, TN=3, FN=7

In other words, the system never triggers stimulation. This leads to:

- **TPR (Sensitivity) = 0%**
- **TNR (Specificity) = 100%**
- **FNR = 100%**
- **Accuracy = 30%**
- **Precision = undefined (no positive decisions)**
- **F1 Score = 0**

Although specificity is perfect, this configuration is clinically unacceptable: all seizure cases are missing. The system behaves in an overly conservative way and fails to provide protection in critical situations

Results after learning

After learning, the final decisions for the same ten scenarios are:

- **TP = 5, FP = 2, TN = 3, FN = 0**

The classical metrics in this case are:

- **TPR (Sensitivity) = 100%**
- **FPR = 40%**
- **FNR = 0%**
- **Accuracy = 80%**
- **Precision = 71.43%**
- **F1 Score ≈ 83.33%**

These values are summarized in Table 8 for the "after learning" configuration.

The adaptive metrics, computed from the change in decisions between the initial and final configurations, are as follows:

- **CR = 70%** → in 7 out of 10 scenarios, the final decision differs from the initial decision, showing that the system is sufficiently flexible and does not simply repeat its initial policy.
- **FCR = 20%** → in 2 out of 10 scenarios, the change led to an incorrect final decision, indicating that most corrections are beneficial.
- **CC = 100%** → all initially incorrect decisions (in particular, all seizure cases) were successfully corrected to match the expert's opinion.
- **IR = 70%** → stimulation was applied in 7 out of 10 scenarios, which reflects an intentionally aggressive strategy toward avoiding missed seizures.

The numerical values of these adaptive metrics are also reported in Table 9. Overall, the results show that learning substantially improves the behavior of the system: all seizure scenarios are now correctly detected, all initial misclassifications are corrected, and the number of unnecessary stimulations, while non-zero, remains at a level that can be further tuned. The remaining false positives suggest that additional refinement is still required to optimize the balance between sensitivity and specificity, but the current behavior is already much closer to the safety requirements expected from an RNS device.

A review of the existing literature indicates that no method or framework has been reported focuses specifically on runtime monitoring and validation of RNS devices using a combination of formal modelling and adaptive decision making. Existing studies mainly address clinical analysis, long-term therapeutic outcomes, or optimization of stimulation parameters, and do not provide an integrated formal and learning-based framework for online validation of device behavior. Thus, a direct comparison of the proposed method with an equivalent approach in the RNS domain is not possible, and the evaluation presented here is based on these ten simulated scenarios and comparison with the initial, non-learning configuration. Within this context, the proposed method is, to the best of our knowledge, the first documented attempt to provide an integrated model based on timed colored Petri nets, fuzzy logic, and reinforcement learning for monitoring and validating RNS devices at runtime.

Table 9 - Evaluation Results for 10 Scenarios

Metric	After Learning (%)
TPR (Sensitivity)	100
FPR (False Positive Rate)	40.00
FNR (False Negative Rate)	0
Accuracy (Overall Accuracy)	80
Precision	71.43
F1 Score	83.33
Correction Rate (CR)	70.00
Correction Coverage (CC)	100.00
False Correction Rate (FCR)	20.00
Intervention Rate (IR)	70.00

6.1. Hierarchical RL Training on the Augmented Scenario Dataset

In addition to the ten expert-defined scenarios used for the baseline evaluation, the behavior of the high-level manager was further analyses on an augmented scenario dataset to study its dynamics and generalization capability. This dataset was constructed from 10 base scenarios with explicit physiological profiles, to which two types of synthetic variations were added: 1140 noisy scenarios, obtained by adding small perturbations to the brain and physiological sensor values, and 360 interpolated scenarios, generated by interpolating between existing profiles. In total, 1530 scenarios were obtained and stored in the file data/scenarios_augmented.csv. This augmented dataset was then used as the training set for the hierarchical manager, while the low-level worker policy was kept fixed and loaded from the pre-trained model (models/worker.pt).

Training was performed for 1500 episodes on this augmented dataset. For each episode, the cumulative return and a moving average over the last 100 episodes were recorded. The training reward curve in Figure 10 shows that after an initial transient phase the learning process stabilizes: the 100-episode moving average of the return fluctuates in a relatively narrow band, typically between approximately 100 and 170, with several episodes achieving very high rewards (up to 485) and a few episodes receiving negative returns. At the end of training (episode 1500), the moving average over the last 100 episodes converges around a value of about 139, indicating that the manager has reached a reasonably stable policy without signs of divergence or collapse.

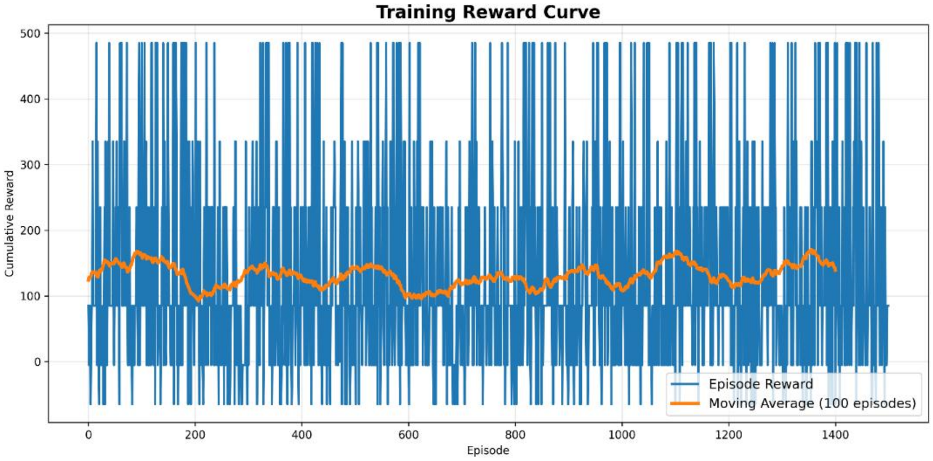


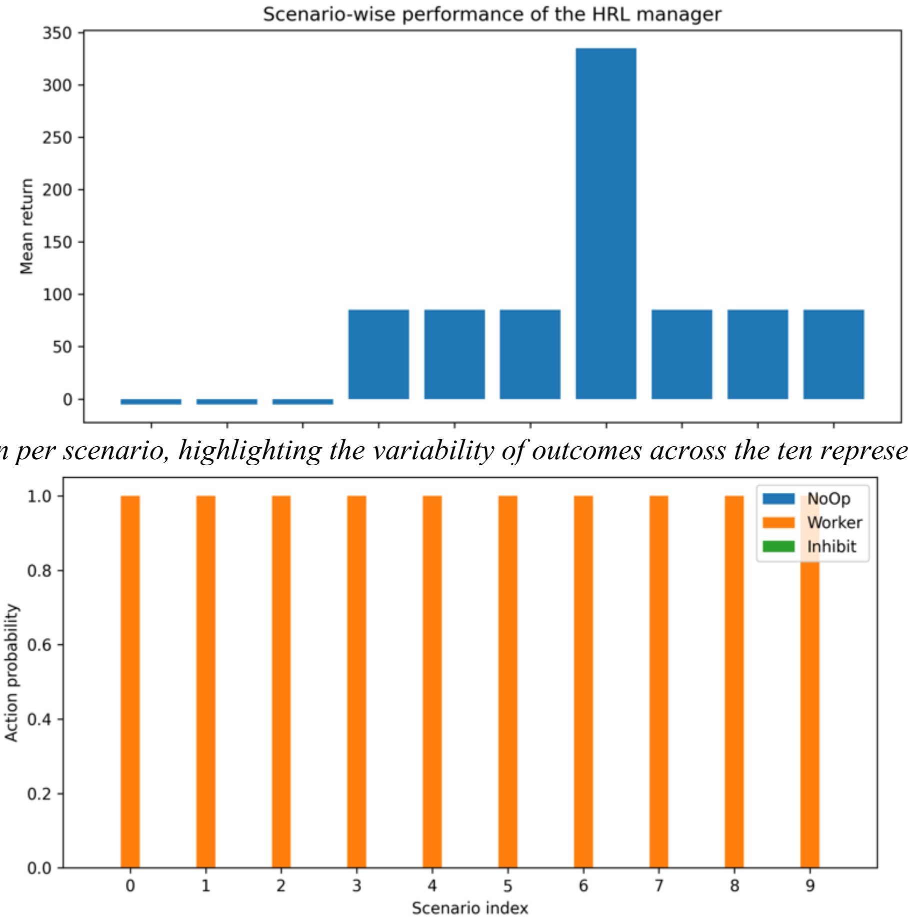
Figure 10- Training reward curve for the hierarchical manager on the augmented scenario dataset

To complement this global view, a scenario-wise evaluation was conducted. For ten representative scenarios (indexed 0–9), multiple rollouts were generated with the trained manager, and for each scenario three quantities were calculated: the mean episode return, the mean episode length, and the empirical probabilities of the three abstract actions (NoOp, delegate to worker, inhibit). In brief:

- **Scenarios 0–2** exhibit a mean return of approximately –5.30 with an average episode length of 1 time step, and the manager selects the worker action with probability 1.0 ([NoOp = 0.00, Worker = 1.00, Inhibit = 0.00]). These cases correspond to conditions where immediate delegation to stimulation leads to a slightly negative outcome, and the episode ends quickly.
- **Scenarios 3–5 and 7–9** achieve a mean return of 85.00 with an average episode length of 5-time steps, again with the manager consistently choosing the worker option ([NoOp = 0.00, Worker = 1.00, Inhibit = 0.00]). Here, repeated delegation to the stimulation worker over several steps is rewarded as beneficial.
- **Scenario 6** yields the best performance, with a mean return of 335.00, an average length of 5-time steps, and the same deterministic choice of the worker action. This indicates a particularly favorable alignment between the scenario profile, the worker's stimulation pattern, and the reward structure.

These results show that, under the current reward design, the manager has learned a consistent and stable policy that heavily favors the stimulation option: in all evaluated scenarios it selects the worker action with probability one. Nevertheless, the variation in mean returns between scenarios (from slightly negative values around –5.30 up to 335) demonstrates that the learned policy is still sensitive to scenario characteristics and that the reward function differentiates clearly between more desirable and less desirable outcomes, even when the high-level action is the same.

This scenario-wise analysis shows (See Figure 11) that, under the reward structure defined in this study, the hierarchical manager agent has converted to a stable and consistent policy that, in the examined scenarios, selects delegation to the worker as its dominant strategy. At the same time, the substantial differences in mean return across scenarios (ranging from negative values to very high values) indicate that the reward function and the learned policy are sensitive to scenario characteristics and do not impose identical behavior on all states. Taking them together, these results on the one hand confirm the model's ability to learn a robust policy for critical scenarios, and on the other hand provide the opportunity to design alternative reward functions and more conservative or less aggressive policies in future studies, without undermining the overall structure of the model (HTCPN + fuzzy logic + hierarchical reinforcement learning).



(a) *Mean return per scenario, highlighting the variability of outcomes across the ten representative scenarios.*

(b) Action selection probabilities for the three abstract actions (NoOp, Worker, Inhibit) in each scenario, showing that under the current reward design the manager consistently delegates to the worker while still producing different returns depending on scenario characteristics.

Figure 11. Scenario-wise performance of the hierarchical manager after training.

7. Conclusion

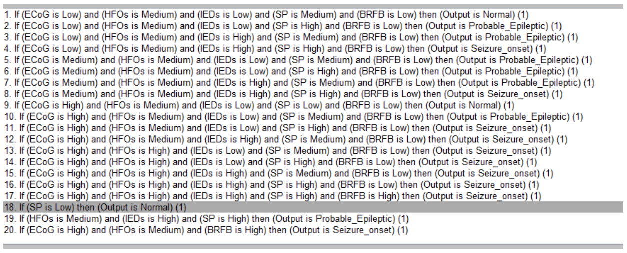
Modern medical systems are gradually moving away from passive, fixed-function operations toward intelligent, adaptive systems with advanced decision-making and personalization capabilities. In this work, the RNS device, as a key tool for epilepsy management, was modelled as a runtime execution framework using timed colored Petri nets combined with fuzzy logic and hierarchical reinforcement learning. The HTCPN layer formally captures the temporal and concurrent behavior of the device, the fuzzy rules encode uncertainty in brain signals and physiological conditions, and the hierarchical RL agent adapts the decision-making process online based on feedback derived from the patient's body state.

Using ten expert-defined scenarios for RNS validation, together with the proposed evaluation metrics, the results show that the model can effectively correct initial misclassifications and accurately identify seizure cases. After learning, the system achieves a recall (TPR) of 100% and a correction coverage (CC) of 100% on these scenarios, meaning that all seizure events are detected and all initially incorrect decisions are successfully corrected to match expert opinion, while maintaining an overall accuracy of 80%. Furthermore, training the high-level manager on an augmented dataset of 1530 scenarios with noisy and interpolated physiological profiles demonstrates that the learned policy remains stable and achieves consistently high returns across a wider range of conditions.

These findings highlight the potential of integrating timed colored Petri nets with fuzzy logic and reinforcement learning for precise and runtime-oriented validation of medical devices such as RNS. Nevertheless, the present evaluation is based on simulated and synthetically augmented scenarios. To further substantiate the clinical relevance and robustness of the approach, future work should consider experiments on larger and more heterogeneous datasets (e.g., PhysioNet and other real-world EEG and physiological databases), as well as more diverse scenarios that explicitly incorporate noise, artefacts, and clinical decision errors. Such studies, possibly combined with hardware-in-the-loop or bench-top prototypes, can provide a stronger basis for assessing the stability, safety, and acceptability of the proposed model in realistic clinical environments.

Appendix

Appendix 1 - Fuzzy Rules



```
1. If (ECoG is Low) and (HFOs is Medium) and (IEDs is Low) and (SP is Medium) and (BRFB is Low) then (Output is Normal) (1)
2. If (ECoG is Low) and (HFOs is Medium) and (IEDs is Low) and (SP is High) and (BRFB is Low) then (Output is Probable_Epileptic) (1)
3. If (ECoG is Low) and (HFOs is Medium) and (IEDs is High) and (SP is Medium) and (BRFB is Low) then (Output is Probable_Epileptic) (1)
4. If (ECoG is Low) and (HFOs is Medium) and (IEDs is High) and (SP is High) and (BRFB is Low) then (Output is Seizure_onset) (1)
5. If (ECoG is Medium) and (HFOs is Medium) and (IEDs is Low) and (SP is Medium) and (BRFB is Low) then (Output is Probable_Epileptic) (1)
6. If (ECoG is Medium) and (HFOs is Medium) and (IEDs is Low) and (SP is High) and (BRFB is Low) then (Output is Probable_Epileptic) (1)
7. If (ECoG is Medium) and (HFOs is Medium) and (IEDs is High) and (SP is Medium) and (BRFB is Low) then (Output is Probable_Epileptic) (1)
8. If (ECoG is Medium) and (HFOs is Medium) and (IEDs is High) and (SP is High) and (BRFB is Low) then (Output is Seizure_onset) (1)
9. If (ECoG is High) and (HFOs is Medium) and (IEDs is Low) and (SP is Low) and (BRFB is Low) then (Output is Normal) (1)
10. If (ECoG is High) and (HFOs is Medium) and (IEDs is Low) and (SP is Medium) and (BRFB is Low) then (Output is Probable_Epileptic) (1)
11. If (ECoG is High) and (HFOs is Medium) and (IEDs is Low) and (SP is High) and (BRFB is Low) then (Output is Seizure_onset) (1)
12. If (ECoG is High) and (HFOs is Medium) and (IEDs is High) and (SP is Medium) and (BRFB is Low) then (Output is Seizure_onset) (1)
13. If (ECoG is High) and (HFOs is High) and (IEDs is Low) and (SP is Medium) and (BRFB is Low) then (Output is Seizure_onset) (1)
14. If (ECoG is High) and (HFOs is High) and (IEDs is Low) and (SP is High) and (BRFB is Low) then (Output is Seizure_onset) (1)
15. If (ECoG is High) and (HFOs is High) and (IEDs is High) and (SP is Medium) and (BRFB is Low) then (Output is Seizure_onset) (1)
16. If (ECoG is High) and (HFOs is High) and (IEDs is High) and (SP is High) and (BRFB is Low) then (Output is Seizure_onset) (1)
17. If (ECoG is High) and (HFOs is High) and (IEDs is High) and (SP is High) and (BRFB is High) then (Output is Seizure_onset) (1)
18. If (SP is Low) then (Output is Normal) (1)
19. If (HFOs is Medium) and (IEDs is High) and (SP is High) then (Output is Probable_Epileptic) (1)
20. If (ECoG is High) and (HFOs is Medium) and (BRFB is High) then (Output is Seizure_onset) (1)
```

Appendix 2 - Data Used in the Ten Scenarios

Sample Brain Sensor Data File

```
95.0 285.0 18.0 0.18 44.0
40.0 80.0 5.0 0.0106 18.0
20 100 5 0.13 10
60 150 8 0.1 25
90 280 18 0.12 45
60.0 150.0 8.0 0.1 25.0
70.0 250.0 15.0 0.14 40.0
30.0 60.0 3.0 0.04 15.0
20.0 35.0 1.0 0.01 10.0
100.0 300.0 20.0 1.0 50.0
```

Sample Physiological Sensor Data File

```
'N' 130.0 150.0 38.5
'N' 110.0 135.0 38.2
'N' 90.0 120.0 37.8
'N' 140.0 150.0 38.5
'H' 101.0 121.0 37.6
'N' 90.0 120.0 37.5
'H' 140.0 160.0 38.7
'H' 100.0 125.0 38.0
'N' 90.0 110.0 37.6
'N' 180.0 150.0 38.5
```

```
'N' 150.0 140.0 38.0
```

Sample Expert Desired Output File

```
Stimulation
Monitoring
Monitoring
Monitoring
Stimulation
Monitoring
Stimulation
Stimulation
Monitoring
Stimulation
```

References

1. Aronson, J.K., C. Heneghan, and R.E. Ferner, *Medical Devices: Definition, Classification, and Regulatory Implications.* Drug Safety, 2019. **43**(2): p. 83-93.
2. Zhang, M., A. Raghunathan, and N.K. Jha, *MedMon: securing medical devices through wireless monitoring and anomaly detection.* IEEE Trans Biomed Circuits Syst, 2013. **7**(6): p. 871-81.
3. Zhang, M., A. Raghunathan, and N.K. Jha, *Trustworthiness of Medical Devices and Body Area Networks.* Proceedings of the IEEE, 2014. **102**(8): p. 1174-1188.
4. Sobrinho, A., et al. *Towards medical device certification: A colored petri nets model of a surface electrocardiography device*. in *IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society*. 2014. IEEE.
5. Sloka Iyengar PhD and Patricia O. Shafer RN, M. *The RNS® System is manufactured by NeuroPace, Inc. Additional information for patients and physicians*. 2017; Available from: https://www.neuropace.com/patients/neuropace-rns-system/.
6. Jensen, K., L.M. Kristensen, and L. Wells, *Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems.* International Journal on Software Tools for Technology Transfer, 2007. **9**(3): p. 213-254.
7. Sutton, R.S., D. Precup, and S. Singh, *Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning.* Artificial intelligence, 1999. **112**(1-2): p. 181-211.
8. Barto, A.G. and S. Mahadevan, *Recent advances in hierarchical reinforcement learning.* Discrete event dynamic systems, 2003. **13**(4): p. 341-379.
9. Dietterich, T.G., *Hierarchical reinforcement learning with the MAXQ value function decomposition.* Journal of artificial intelligence research, 2000. **13**: p. 227-303.
10. Kulkarni, T.D., et al., *Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation.* Advances in neural information processing systems, 2016. **29**.
11. Pateria, S., et al., *Hierarchical reinforcement learning: A comprehensive survey.* ACM Computing Surveys (CSUR), 2021. **54**(5): p. 1-35.
12. Zadeh, L.A., *Fuzzy sets.* Information and control, 1965. **8**(3): p. 338-353.
13. Majma, N., S.M. Babamir, and A. Monadjemi, *Runtime Verification of Pacemaker Functionality Using Hierarchical Fuzzy Colored Petri-nets.* Journal of Medical Systems, 2016. **41**(2).
14. Liu, Q., K.G. Mkongwa, and C. Zhang, *Performance issues in wireless body area networks for the healthcare application: a survey and future prospects.* SN Applied Sciences, 2021. **3**(2).
15. Narwal, B. and A.K. Mohapatra, *A survey on security and authentication in wireless body area networks.* Journal of Systems Architecture, 2021. **113**.
16. Alzahrani, B.A., et al., *A Provably Secure and Lightweight Patient-Healthcare Authentication Protocol in Wireless Body Area Networks.* Wireless Personal Communications, 2020. **117**(1): p. 47-69.
17. Yaacoub, J.-P.A., et al., *Securing internet of medical things systems: Limitations, issues and recommendations.* Future Generation Computer Systems, 2020. **105**: p. 581-606.
18. Ali, S., et al., *Towards Pattern-Based Change Verification Framework for Cloud-Enabled Healthcare Component-Based.* IEEE Access, 2020. **8**: p. 148007-148020.
19. Assiri, M., *Modeling Cardiac Pacemakers With Timed Coloured Petri Nets And Related Tools*. 2021.
20. Al Hamadi, H., A. Gawanmeh, and M. Al-Qutayri. *A verification methodology for a wireless body sensor network functionality*. in *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. 2014. IEEE.
21. Majma, N. and S.M. Babamir, *Model-Based Monitoring and Adaptation of Pacemaker Behavior Using Hierarchical Fuzzy Colored Petri-Nets.* IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020. **50**(9): p. 3344-3357.
22. Majma, N. and S.M. Babamir, *Modeling and Simulation of Fuzzy-rule Based WBAN using Multi-level Fuzzy Colored Petri-nets and Reinforcement Learning.* Journal of Computational Science, 2024: p. 102455.
23. Haur, I., J.-L. Béchennec, and O.H. Roux, *Model-Checking of Concurrent Real-Time Software Using High-Level Colored Time Petri Nets with Stopwatches.* Cybernetics and Systems, 2023: p. 1-31.
24. Júnior, J.I.F., et al., *A coloured Petri nets-based system for validation of biomedical signal acquisition devices.* The Journal of Supercomputing, 2024: p. 1-30.
25. Frommeyer, T.C., et al. *Reinforcement learning and its clinical applications within healthcare: A systematic review of precision medicine and dynamic treatment regimes*. in *Healthcare*. 2025. MDPI.
26. Jayaraman, P., et al., *A primer on reinforcement learning in medicine for clinicians.* NPJ Digital Medicine, 2024. **7**(1): p. 337.

27. Liu, S., et al., *Reinforcement learning for clinical decision support in critical care: comprehensive review.* Journal of medical Internet research, 2020. **22**(7): p. e18477.
28. Nagaraj, V., A. Lamperski, and T.I. Netoff, *Seizure control in a computational model using a reinforcement learning stimulation paradigm.* International journal of neural systems, 2017. **27**(07): p. 1750012.
29. Pineau, J., et al., *Treating epilepsy via adaptive neurostimulation: a reinforcement learning approach.* International journal of neural systems, 2009. **19**(04): p. 227-240.
30. Dan, R., H. Zhang, and J. Bai, *Closed-Loop Control of Epilepsy Based on Reinforcement Learning.* International journal of neural systems, 2025: p. 2550074.
31. Gao, Q., et al. *Offline learning of closed-loop deep brain stimulation controllers for parkinson disease treatment*. in *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*. 2023.
32. Zhong, C., et al., *Hierarchical reinforcement learning for automatic disease diagnosis.* Bioinformatics, 2022. **38**(16): p. 3995-4001.
33. Hao, Q., et al. *Hierarchical reinforcement learning for scarce medical resource allocation with imperfect information*. in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 2021.
34. Li, S. and X. Sun, *Dialogue-Based Disease Diagnosis Using Hierarchical Reinforcement Learning with Multi-Expert Feedback.* International Journal of Advanced Computer Science & Applications, 2025. **16**(2).