IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 23, 2019, accepted December 5, 2019, date of publication December 10, 2019, date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2958827

Brain-Inspired Intelligence for Real-Time Health Situation Understanding in Smart e-Health Home Applications

MAHDI NAGHSHVARIANJAHROMI, (Student Member, IEEE), SHIVA KUMAR, (Member, IEEE), AND M. JAMAL DEEN, (Fellow, IEEE)
Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada

Corresponding author: M. Jamal Deen (jamal@mail.ece.mcmaster.ca)

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC), and in part by the Canada Research Chair Program (CRCP).

ABSTRACT The autonomic computing layer of the smart e-Health home based on a cognitive dynamic system (CDS) can be a solution for improving health situation understanding, reducing the healthcare system costs, and improving people's quality of life. It can also be a solution for reducing the large number of sudden deaths outside of a hospital due to fatal diseases such as Arrhythmia. Towards this objective, we start from understanding the health situation, by diagnosing healthy and unhealthy persons. For this, we developed a decision-making system that is inspired by the medical doctors (MDs) decision-making processes. Our system is based on a CDS for cognitive decision-making and it can create a decision-making tree automatically. The simple, low complexity algorithmic design of the proposed system makes it suitable for real-time applications. A proof-of-concept case study of the implementation of the CDS was done on Arrhythmia disease. An accuracy of 95.4% was achieved using the proposed algorithms. Also, these algorithms can make a decision in less than 80 ms, and for one User, this includes the time for training. The proposed platform can be extended for more healthcare applications such as screening, disease class diagnosis, prevention, treatment, or monitoring healing. As a result, the proposed CDS algorithms can be an example of the first step for designing the autonomic computing layer of a smart e-Health home platform.

INDEX TERMS Autonomic decision-making system, autonomic computing layer, cognitive dynamic system, cognitive decision making, non-Gaussian and non-linear environment, situation understanding, smart systems, smart e-health home, autonomic decision-making system, medical doctor decision making, cognitive dynamic system.

I. INTRODUCTION

Currently, the autonomic decision-making systems (ADMS) [1] for smart interactive cyber-physical systems (CPS) are attracting much attention from researchers and technology providers [1]–[6], [27]. In this paper, a cognitive dynamic system (CDS) based on the ADMS concept is presented for diagnostic tests in a smart e-Health home. The CDS is inspired by the neuroscience model of the human brain and it is built on the principles of cognition, i.e., perception-action cycle (PAC), memory, attention, intelligence and language [2]–[4]. CDS has found applications in the smart home [2] and long-haul fiber-optic links [4]–[6]. CDS is proposed as an alternative to artificial intelligence (AI) for many AI applications [2]–[6]. Also, CDS creates internal rewards and then uses the rewards to take some actions. However, AI takes actions based on the rewards from the environment (see Section VI.A.2). Therefore, in designing the autonomic decision layer in a smart e-Health home, we choose CDS instead of AI.

The algorithmic presentation of a CDS for linear and Gaussian environments (LGE) were presented in [3]. In [4]–[6], a CDS was proposed for smart fiber optic communication systems as an example of using CDS for decision making in complex smart systems. It should be mentioned that we use the term cognitive decision making (CDM) for decision making using CDS. Also, complex smart e-Health systems such as smart e-Health home are non-Gaussian and nonlinear health

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

environments (NGNLHE). Here, NGNLHE means that the outputs of the healthcare system are not linearly dependent on the input. Also, the outputs of healthcare systems do not have Gaussian distributions.

The CDS was presented for a smart home in [2] as an enhanced-AI. While a CDS block diagram was presented, no detailed algorithms neither discussed nor provided. It should be noted that, the smart home is different from a smart e-Health home in terms of applications. Therefore, in this paper, we propose the detailed algorithms of a CDS for diagnostic tests in a smart e-Health home.

The basic model of a CDS is provided in Fig. 1. Besides, Fig. 1 shows the three main subsystems in a CDS: (i) Perception by the perceptor; (ii) a Feedback channel for sending the internal rewards; and (iii) the Executive for performing actions on the environment, which together create the PAC. More details about CDS and PAC can be found in [3]. In this paper, CDS is used as the subsystem of ADMS for semi-medical diagnostic applications with a policy for low false alarm rates.

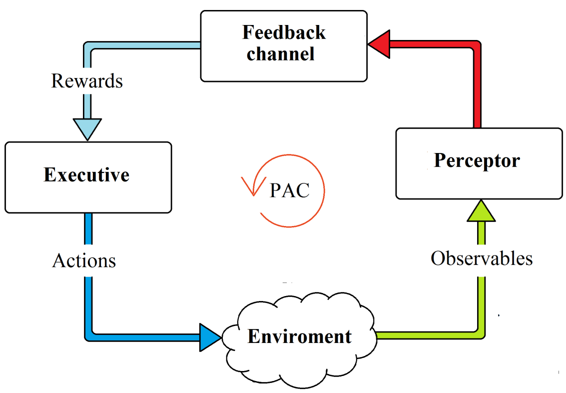


FIGURE 1. Block diagram of a cognitive dynamic system (CDS). PAC: Perception action cycle.

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

1. The concept of CDS is used as an alternative to AI in ADMS and for the non-Gaussian and non-linear health environment (NGNLHE). The inspiration of CDS from natural human intelligence brings the advantage to simulate a medical doctor (MD) decision-making approach with less complexity.
2. The architectural structure of an ADMS is designed as the first step of the health situation understanding and diagnosis. In this paper, we focus on the diagnostic tests between a healthy and unhealthy situation using ADMS towards the goal of low false alarm policy.
3. Algorithms for decision making between healthy and unhealthy situations are presented. These algorithms are designed based on creating a decision-making tree in the perceptor. Also, the algorithms use a semi-medical doctors' one-dimensional decision-making approach. Therefore, the CDS decision making is based on only one feature. Also, in the proposed algorithms, measurements using actuators and sensors are regarded as cognitive actions. The decision based on one feature results in less complexity and simple algorithms for real-time decision-making in a NGNLHE. Further, low complexity and simple algorithms will help in designing a platform for other healthcare policies such as screening or disease class diagnosis. Because we are using low complexity and straightforward algorithms, the system can be trained dynamically and makes a decision quickly. As a result, the CDS is applicable for many real-time applications in a smart e-Health home.
4. We have designed a flexible ADMS to provide acceptable accuracy in different medical situations. For example, if the database for training the system is insufficient, then the system acts like MDs. Therefore, for known User-health conditions, the ADMS can determine if someone is healthy or unhealthy by extracting the normal range of medical features. As another example, in the case of some sensors failing, the ADMS can find alternative features from the other available sensors. These alternative features will lead to new actions to give the best possible diagnosis. For example, if the system faces a blood pressure sensor failure, then it may check the lung sound and heart rate instead of blood pressure.
5. A proof-of-concept case study in which a virtual CDS-based ADMS is applied to Arrhythmia diagnosis is performed. It is shown that this new design performs with 95.4% accuracy. This accuracy is achieved even with missing measured information for some features such as the heart rate of some Users. These missing features can be considered as sensors' failures. In Table 1, a comparison between the proposed method and some related works on this Arrhythmia database decision-making accuracy is shown.

TABLE 1. Comparison between proposed work and related published works.

References	Technique implemented	Best reported accuracy (%)	Sensitivity (Diagnosis abnormal rhythm accuracy) (%)	Specificity (Normal rhythms accuracy) (%)
[7]	Random forest (RF)+Support vector machine (SVM)	77.4	59.9	91.4
[8]	Deep learning	75.8	-	-
[9]	SVM for 2 classes and 11 features	86	-	-
This Work	ADMS using CDS	95.4	90	100

This paper is organized as follows. In Section II, related works are briefly discussed. In Section III, a brief description of why we use the CDS for the diagnostic tests is given. In Section IV, the autonomic computing layer of a smart e-Health home is presented. In Section V, non-Gaussian and nonlinear health environments are discussed. In Section VI, the architectural structure of CDS for the application of ADMS for NGNLHE and the proposed algorithms for zero false alarm health condition diagnosis are discussed. In Section VII, we discuss the simulation results for the proof-of-concept case study for Arrhythmia diagnosis. In Section VIII, we present the research challenges and future work. Finally, the conclusions are presented in Section IX.

II. RELATED WORKS

Here, we provide a very brief review of related published methods using machine learning or AI techniques. These techniques are applied for diagnostic testing or decision making in healthcare applications. Recently, because of the increasing popularity of wearable and portable health sensors, health records and data have rapidly grown. As a result, a large number of healthcare datasets have been generated [10]. To predict clinical outcomes or clinical problems, clinicians and researchers apply machine learning and AI algorithms using available datasets [11]. Also, machine learning is being applied to the diagnosis of various diseases [12]–[15]. Besides, these techniques can also be used for disease treatment and optimal decision making [16]–[18].

In [19], machine learning was used to reduces false Arrhythmia alarm from electrocardiogram (ECG) signals. In [20], [21], it was shown that machine learning can even be used to generate reports from medical images [20], [21]. Here, the generalizable prediction models can be extracted by machine learning-based approaches. Then, the patterns of the measured data can be extracted. Thus, extracted patterns help MDs to perform more personalized clinical prediction in patients [22].

III. WHY A COGNITIVE DYNAMIC SYSTEM?

In this section, we briefly discuss machine learning scenarios. Then, we discuss how CDS is selected among popular machine learning scenarios. In this paper, AI uses machine learning approaches to creates intelligent machines. AI created by machine learning is different from symbolic rule-based AI. Therefore, instead of programming predefined rules, AI using machine learning can learn from datasets, examples and experiences. In machine learning-based AI, a machine learning algorithm extracts the model from the dataset. Then, the model can be used for prediction. Also, the algorithm can learn to optimize models based on dataset and policy (special task such as low false alarm policy diagnostic test).

A. MACHINE LEARNING APPROACHES

Machine learning is an interdisciplinarity field in computer science, mathematics and statistics. Generally, there are many machine learning approaches such as supervised learning, reinforcement learning, semi-supervised learning, unsupervised learning and transfer learning. We will only focus on the first two main types supervised learning and reinforcement learning.

1) SUPERVISED LEARNING

The most popular approach in machine learning for practical applications such as predicting the length of stay in hospital, medication response and health condition is supervised learning (SL). SL can find patterns inside data. In general, the SL algorithm can learn how to create a classifier for predicting the output variable y for a given input variable x. The SL algorithm extracts a mapping function f where $y = f(x)$. An algorithm with a set of data $\{x_1, x_2, \ldots, x_n\}$ with the corresponding output label $\{y_1, y_2, ..., y_n\}$ builds the classifier.

SL can be divided into two main branches: (1) learning by type of prediction and (2) learning by type of modeling. The type of prediction problems can be divided into regression or classification. For predicting continuous output data, regression learning methods can be more suitable. For the prediction of output class, classification algorithms such as logistic regression, naive Bayes, decision trees, or support vector machines (SVM) are better choices [23]. For example, a child's height prediction can be made better by linear regression. However, the decision tree or naive Bayes are better for binary diagnostic test prediction. The type of modeling can be the extraction of a discriminative model such as decision trees and SVM algorithms. These algorithms can extract the decision boundary within the data based on the learning goal. However, machine learning methods such as naive Bayes or Bayesian approaches, can learn the probability distributions of the data.

In summary, SL trains an algorithm on a labeled database to predict the correct outputs for the unseen inputs. SL can be applied to problems with input/output. Also, the labeled dataset should be available for SL. In addition, SL can be used for prediction and classification, such as image recognition and filtering spam emails.

2) REINFORCEMENT LEARNING

Figure 2 shows a schematic diagram of reinforcement learning (RL). RL maps a decision-making problem into an interaction of a computer agent with a dynamic environment by trial and error [24]. The computer agent attempts to reach the best reward based on feedback received from the dynamic environment when it searches the state and action spaces. For example, in healthcare applications, the RL algorithm will try to improve the model parameters based on iteratively simulating the state (User health condition). Then, after applying the action (e.g., activating or deactivating sensors, amount of medication delivery and modeling accuracy), the agent obtains the feedback reward from the environment (healthy or unhealthy approval by MDs in the clinic). The RL algorithm can then converge to a model that may generate optimal decisions [16].

In summary, RL learns through trial and error from interaction with a dynamic environment such as learning to play

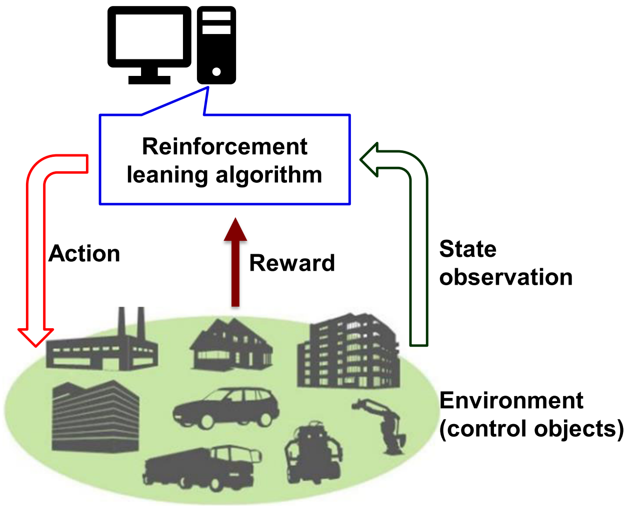


FIGURE 2. Schematic of reinforcement learning.

a game or a movie/video recommendation system. There are states and actions in RL. Typically, no database is required for RL and it can find the action for optimizing the reward. RL receives the reward/punishment from the dynamic environment.

3) PROPOSED COGNITIVE DYNAMIC SYSTEM

Figure 3 shows the conceptual implementation of our proposed CDS for the healthcare environment (See Fig. 10 later). By combining SL and RL as two main scenarios of machine learning, CDS can be considered as an enhanced AI. The perceptor of the CDS can extract the model using a SL algorithm. Using the extracted model, the perceptor can generate the internal reward and predict the dynamic environment outcome [3]. In this paper, dynamic environment is the health environment. Perceptor sends the internal reward to the executive through the feedback channel. RL in the executive uses the internal reward in the current PAC to find an action that can optimize the internal reward for the next PAC.

In summary, unlike typical RL, the executive of the CDS can use the extracted model by the perceptor to apply a cognitive action on the dynamic environment. The internal reward gives the CDS self-awareness, self-consciousness and independence from the dynamic environment. Briefly, the CDS has the *"conscience"* about the action. For extracting the internal reward, the SL algorithm should be applied in the perceptor. However, in typical RL, the agent applies the actions blindly to receive feedback from the environment. Therefore, the CDS is a more appropriate choice rather than a typical RL in intelligent machine applications.

IV. SMART E-HEALTH HOME BASED ON CYBER-PHYSICAL SYSTEM ARCHITECTURE

Globally, Internet-of-Things (IoT) is attracting much attention from researchers and technology developers and providers [25], [26]. IoT technology is based on connecting a variety of conventional devices and systems such as sensors, actuators, appliances, TV and cars with computing devices to have the capability to automatically transfer data over a network. Therefore, IoT creates a network of intelligent systems that can communicate with each other or with human Users [25], [26]. In recent years, the considerable advances in computing, wireless and network communications among and from low cost and low power sensors, actuators, and electronic components have made many previous fictional applications of IoT now practical, an example being the smart e-Health home presented in [27]. This application can potentially make our lives more comfortable and safer, as well as dramatically reduce the healthcare system cost for elderly healthcare. For simplicity, we focused on the cyber-physical system (CPS) sub-area of IoT for the architecture of a smart e-Health home. Fig. 4 shows the smart e-Health home architecture as a cyber-physical system. A smart e-Health home can be considered to have four main layers [27]:

- **Sensors and Actuators:** These are environmental and medical sensors, appliance and home control units, or any device or methods providing information between Users (humans) and computers/robots

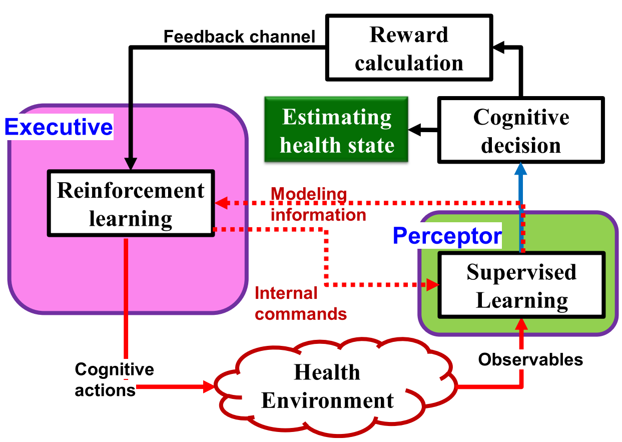


FIGURE 3. Conceptual implementation of proposed CDS.

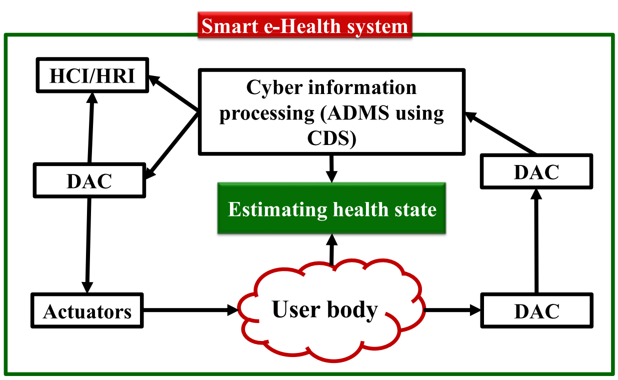


FIGURE 4. Smart e-Health home architecture with a autonomic decision-making system (ADMS) using cognitive dynamic system (CDS) as the cyber physical system (CPS). ADC: Analog-to-digital converter, DAC: Digital to analog converter, HCI: Human-computer interaction, HRI: Human-robot interaction.

in human-computer interaction (HCI) or human-robot interaction (HRI).

- **Home communication network:** This layer is responsible for information gathering and management, feature extracted from signals and discovery of appliances or sensors.
- **Autonomic computing:** This layer is responsible for knowledge management, situation understanding of the inhabitants such as whether the User has a disease or not, intelligent reasoning and decision making. This cyber part of the smart e-Health home is termed as the ADMS for information processing of the measured signals from the sensors.
- **Safety and healthcare services:** This can include health-care services (hospital/doctors), safety services (police, fireguards), remote support, telemedicine, e-Health networks and other smart e-Health services.

In this paper, we focus on the autonomic computing layer of a smart e-Health home. The implementation bottlenecks of the *autonomic computing layer* of smart e-Health homes were discussed in [28]–[30]. As we mentioned before, the *autonomic computing layer* can be termed as the ADMS [31], [32]. ADMS is responsible for information processing that captures sensory data of the smart home environment as well as those related to the health of the inhabitants (Users). Typically, the ADMS can be implemented using an artificial intelligence (AI) technique. However, in this paper, we implemented the ADMS using the cognitive dynamic system (CDS) concept.

ADMS can be used for different situations or recommendations with minimal human intervention. Note that ADMS is not intended to replace medical doctors, but the ADMS can be inspired from MDs decision-making approaches. Also, it can perform time-consuming tasks, and handle emergency issues in which time is a critical element in saving the life of the User using real-time monitoring at home. Therefore, Users can have health monitoring as needed in a comfortable environment (their home) without interrupting their daily activities.

In this paper, we implemented the proof-of-concept of ADMS using the proposed CDS for diagnostic tests or low false alarm policy. We focus on a low false alarm policy (someone has a disease or no disease) as the first stage of our work. For the implementation of a low false alarm policy, the CDS is inspired by the decision-making approaches of medical doctors (MDs). On the other hand, if a person with an unknown or a new disease visits an MD, then the MD can still make the diagnosis that the person is not healthy. The MD can diagnose the condition based on measured abnormal symptoms or features. Therefore, we choose the binary decision making of a User's health state as the first step in designing an ADMS using CDS.

The binary decision making between healthy and unhealthy persons has a key role in designing the ADMS of a smart e-Health home. This is because binary decision-making results can be used as the base for developing a future comprehensive ADMS of a smart e-Health home such as disease class diagnosis or screening (See Section VIII.B also). Therefore, the proposed CDS can be considered as the first step in designing a simple "cyber semi-medical doctor's decision-making system (CSMDDMS)" for situation understanding between a healthy and unhealthy User in the *autonomic computing layer* of a smart e-Health home.

V. NON-GAUSSIAN AND NONLINEAR HEALTH ENVIROMENT

Often, the data obtained or measured from health conditions are not normally distributed (non-Gaussian) [33], [34]. Also, most of the extracted features of the measured sensors' signals are non-linear functions of the human health condition. Therefore, human health can be considered as a non-Gaussian and non-linear environment, such as breast cancer modeling [34].

The conventional CDS described in [3] is designed and implemented for a linear and Gaussian environment. Therefore, standard CDS cannot apply to a non-Gaussian and non-linear health environment (NGNLHE) for decision making. For decision making on the health condition of the User number u, the relation between extracted features from measured signals by the sensors and the User number u health conditions can be defined as follows:

$$\begin{bmatrix} o_1 \\ o_2 \\ \vdots \\ o_N \end{bmatrix} = \begin{bmatrix} g_1(hd_1, hd_2, \ldots, hd_H, w_1) \\ g_2(hd_1, hd_2, \ldots, hd_H, w_2) \\ \vdots \\ g_N(hd_1, hd_2, \ldots, hd_H, w_N) \end{bmatrix}$$

or

$$O = G(HD, W), \tag{1}$$

where the vector $\boldsymbol{O}$ refers to features that are extracted from measured signals from sensors such as the electrocardiogram (ECG), photoplethysmogram (PPG), or electroencephalogram (EEG), or the typical information such as sex, height and heart rate. $\boldsymbol{HD} = [hd_1, hd_2, \ldots, hd_H]$ refers to the health conditions we would like to diagnose. For example, the $hd_1, hd_{h\neq 1,H}$ and hd_H are User health state without diseases, with disease class of $hd_{h\neq 1,H}$ and with unknown disease class, respectively. $\boldsymbol{W} = [w_1, w_2, \ldots, w_N]$ represents the non-Gaussian noise. The feature o_n is a nonlinear function of $g_n(\cdot)$. Also, a feature o_n has a non-Gaussian distribution for any health condition of hd_h. In (1), for a specific User, the vector $\boldsymbol{O}$ is known from measurements and the User's health condition is unknown. For simplicity, we just consider the binary case to find someone without the diseases h_1 and $hd_{h>1} = \{hd_2, \ldots, hd_H\}$. As we know, medical doctors (MDs) can diagnose between these two conditions (i.e., someone with or without disease) using the measured feature o_n. Therefore, our CDS architecture is inspired by MDs decision-making method in a NGNLHE. In the first step, the CDS is designed to make a decision between a healthy or unhealthy condition.

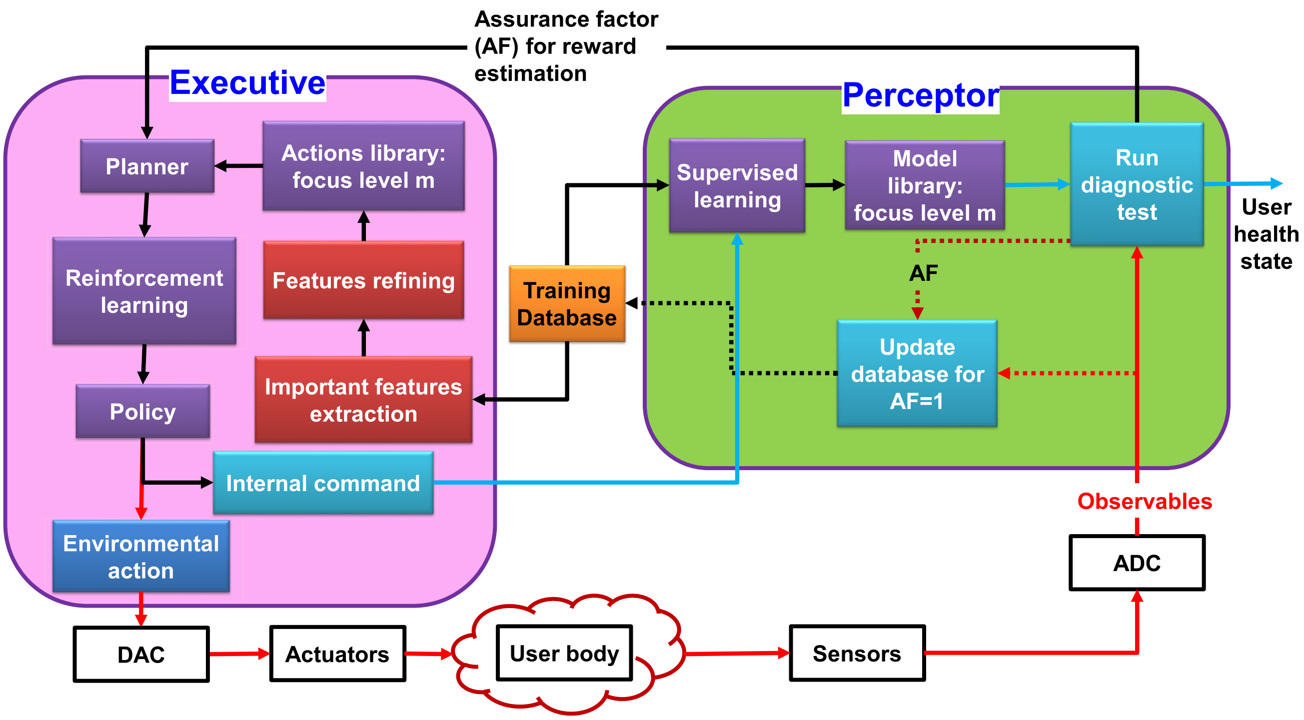


FIGURE 5. Block diagram of proposed CDS architecture for the ADMS of a smart e-Health home.

VI. PROPOSED ADMS USING CDS ARCHITECTURE AND ALGORITHMS

In this section, we describe proposed the CDS architecture and algorithms that can be applied for decision-making purposes in NGNHLE. The detailed proposed CDS architecture for health situation decision making is shown in Fig. 5. A CDS has two main subsystems: (i) the perceptor, and (ii) the executive with a feedback channel linking them. Through interactions with the Users (environment) and e-Health network, a PAC is created in the form of the training and prediction modes, using local feedback loops and a global feedback loop, respectively.

First, we describe the perceptor and executive training modes using related algorithms. Next, we discuss the decision-making tree, the feedback channel and internal rewards in training mode or prediction mode for both the perceptor and the executive. The CDS operates in three modes: (i) training (ii) prediction and (iii) steady state. The CDS functions will be different depending on its mode. The steady-state mode means that nothing seems abnormal, or there is no User request.

In Fig. 5, the training database can be updated dynamically in three ways:

1. The data and information collected by the sensors for Users with a known health state (Users with a known health state, or after 100% assurance decision making depending on the policy and focus level).
2. The database can be updated through e-Health and healthcare networks.
3. The smart e-Health home can be a node in a network of smart e-Health homes.

The data and information collected in the database will be converted to the model library in the perceptor and actions library in the executive after running the CDS in its training mode. The perceptor and executive will gain more knowledge dynamically after the training database is updated (see Fig. 5). The CDS will use the knowledge in the perceptor model library for prediction of the health conditions of Users with unknown health states.

For specific situations or on request by the Users such as something abnormal in the home or crying out with pain, needing help or the User saying directly, "I am not feeling well," then the CDS can initiate the prediction mode. The knowledge is used to reason and predict the health situation of the User, and then to determine whether any action is needed based on the policy and objectives. Here, the policy is with low false alarm in diagnosing between healthy and unhealthy conditions. Knowledge in the perceptor is represented as a set of concepts and the relationships among them, together with the User's health situations.

Therefore, in a specific situation or on a User's request, the CDS will switch from steady-state to the prediction mode state, and the fuzzy nature of this process is presented in Section VI.A below. Also, the CDS in the training mode dynamically updates the knowledge in the perceptor and actions library in the executive after the database is updated in the three ways discussed above. This update can be done in real-time and even in parallel to the prediction mode or during the steady-state mode. In Sections VI.B to VI.D, the algorithmic description for the training and prediction modes are described. Finally, in section VI.E, the complexity of the presented algorithms are discussed.

A. FUZZY NATURE OF THE PROPOSED COGNITIVE DYNAMIC SYSTEM FOR SMART E-HEALTH HOME APPLICATIONS

Typically, for the internal reward calculation, the CDS uses Shannon's theory [3] to calculate the entropic state at time n. The technique based on the *Shannon theory* is applied to a linear and Gaussian environment (LGE). However, in a non-Gaussian and nonlinear health environment (NGNLHE), the complexity of using standard CDS internal reward calculation is higher than in the typical LGE.

For real-time applications in this paper, the algorithms should be simple, straightforward and fast. Thus, we would like to avoid using extra logarithms, complicated functions and integrals required in the entropic state calculation provided by standard CDS. Here, the internal reward is inspired by the fuzzy human decision-making approach with lower computational cost (especially for making a decision in a complex health environment).

Fuzzy logic here means that the logic values of variables can be a real number between 0 and 1 [35]–[37]. Also, fuzzy logic is widely used for medical decision making in health environments such as Value-Laden choices [38], medical decision making in the intensive care unit (ICU) [39], and artrial fibrillation detection [40], [41].

Fuzzy logic can be presented as the assurance about the decision. For example, we can make the wrong decision when the assurance is less than 1. Similarly, the proposed CDS can measure the assurance about the decision after taking actions (See eq. (7) and Section VI.D).

In this paper, the proposed CDS runs a diagnostic test for a User, then diagnoses between the healthy or unhealthy situation with a low false alarm policy. Based on this policy, the CDS assumes that the User is healthy at first, but without taking any actions, the confidence and assurance factor is zero.

As mentioned above, the first assumption/decision of the CDS is that the User is healthy. However, without doing any actions, the system has zero confidence and zero assurance about the decision of the User being healthy. If the system finds that the person is not healthy, this is 100% assurance, because the key feature has a value outside of the normal range, then all processes will be stopped. In this case, the User should be referred for disease diagnosis to the medical doctors or other algorithms. However, by doing more actions, the CDS can be more confident about the first assumption and decision (that is, the User is healthy). In any focus level, if the CDS meets the predefined threshold of assurance (e.g., 97.5% of assurance) or acceptable probability of error (e.g., 2.5% chance of error) about the initial decision that the User is healthy, then the CDS can send the User to the prevention part of the healthcare system.

If the focus level of the CDS cannot be increased, but the CDS finds all key features are within the normal range and the CDS does not assure enough about the health condition of User (e.g., 94% assurance $<$ 97.5% accuracy threshold), then it will claim the User as healthy. However, the CDS knows that the threshold was not met for the User and refers the User to medical doctors for screening or to other screening algorithms. In screening, unlike our proposed algorithms in this paper, the assumption and policy are that the User is unhealthy unless the healthcare system can find evidence that shows that the User is healthy. Furthermore, in screening, a high false alarm is acceptable.

B. HIGH-LEVEL ALGORITHM PRESENTATION FOR THE PROPOSED CDS

In this section, we describe the high-level presentation algorithm for the proposed CDS that can be applied for the ADMS of a smart e-Health home. The high-level algorithm shows the outline of the complete procedure of a proposed CDS in one focus level. Table 2 lists the notations used in this paper for easy referencing.

The algorithm is described briefly as follows (Please see Fig. 5):

- CDS training mode:
  a) Perceptor training mode.
  b) Executive training mode.
- CDS prediction mode:
  a) Executive (planner).
  b) Executive (reinforcement learning).
  c) Perceptor (run diagnostic test).
  d) Perceptor (assurance factor calculation).
  e) Executive (policy).

High-Level Algorithm for Proposed CDS Using Algorithms 1-4

CDS training part: Re-run after updating the database or focus level increasing by internal commands

Perceptor training part (see algorithms 1-3 and Fig. 5):

Creating the decision tree for focus level m

Modeling algorithm extracts the normal ranges for each feature and creates the model library for focus level m

Executive training part:

Extract key features from sensors and disease class of $hd_{h>1} \in \{hd_2, \ldots, hd_H\}$

Important feature refining (has new information about User-health) and creating actions library for focus level m

CDS User-health prediction (See Algorithm 4 and Fig. 5):

1. Depend on User request or periodically test run the diagnostic test in focus level $m = 1$

2. Check if the User has the disease $hd_h \in \{hd_2, \ldots, hd_H\}$

3. **Executive (Planner)**: Create a buffer action space for disease class of hd_h as the $\boldsymbol{A}$ vector

4. **Executive (Reinforcement learning)**: Finds the best action to minimize the reward for disease hd_h using reinforcement learning and remove this action from the buffer action space of $\boldsymbol{A}$
5. **Perceptor:** Run the diagnostic test and load the model for the related feature from the model library
6. **Perceptor (Decision making):** If the User is unhealthy: Alarm and send for the disease diagnosis process
7. **Perceptor (Raw internal reward calculation):** If the User healthy: calculate the assurance factor (AF)
8. **Perceptor:** If the (1-AF) $\leq$ *Threshold*: Claim User is healthy and start healthy living recommendations
9. **Executive (Reinforcement learning):** If the (1-AF) $>$ *Threshold*: Find the next best action that minimizes the reward in the actions buffer space of $\boldsymbol{A}$
10. **Executive (policy):** If there is no more actions buffer space of $\boldsymbol{A}$ and (1-AF) $>$ *Threshold*: Run the CDS prediction process for hd_{h+1} and $h+1 \leq H$
11. **Executive (policy):** If all $hd_h \in \{hd_2, \ldots, hd_H\}$ are checked, and the User does not have any $hd_h \in \{hd_2, \ldots, hd_H\}$ and still (1-AF) $>$ *Threshold* and if focus level can be increased (see eq. 2): Send internal commands to the perceptor for increasing the focus level and re-run all CDS training and prediction modes for new focus level of $(m+1) \leq M$
12. **Executive (policy):** If all $hd_h \in \{hd_2, \ldots, hd_H\}$ are checked, and the User has not any $hd_h \in \{hd_2, \ldots, hd_H\}$ and still (1-AF)>*Threshold*, but focus level cannot be increased (see eq. 2): Claim User as the healthy and send the User for the screening

C. TRAINING MODE: PERCEPTOR AND EXECUTIVE

In a conventional CDS, the perceptor should have *Bayesian filtering* based on the *Kalman filter*. However, the *Kalman filter* cannot be used for non-Gaussian environments. Because, the *filter* equations are extracted for a linear and Gaussian environment [3]. Therefore, the Bayesian filter is omitted, and instead, the Bayesian equation is directly used for extracting the posterior in decision making. The multilayer Bayesian generative model using decision trees is a typical method in machine learning. The CDS training mode can be summarized in the following three parts which can be described as follows:

1. Creating a decision-making tree (see Algorithm 1),
2. Knowledge and actions extraction from the database (see Algorithm 2)
3. Actions refinement in the executive library (see Algorithm 3).

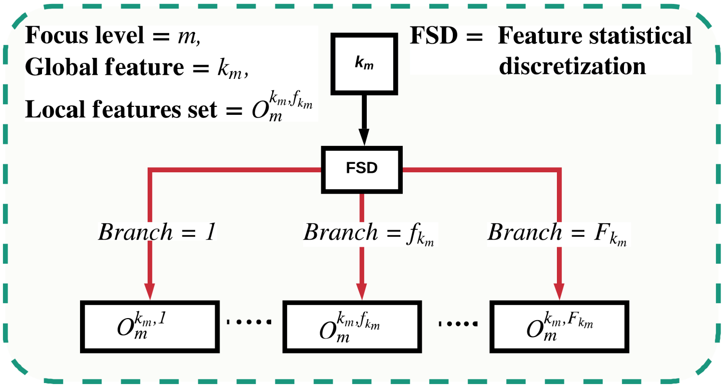
1) CREATING A DECISION TREE

Medical doctors typically use the decision-making tree approach for screening or diagnostics. For example, the pediatric tachycardia algorithm with a pulse and poor perfusion are discussed in [42], [43]. Therefore, the CDS should automatically create this decision-making tree. A general schematic of the proposed decision-making tree is shown in Fig. 6a.

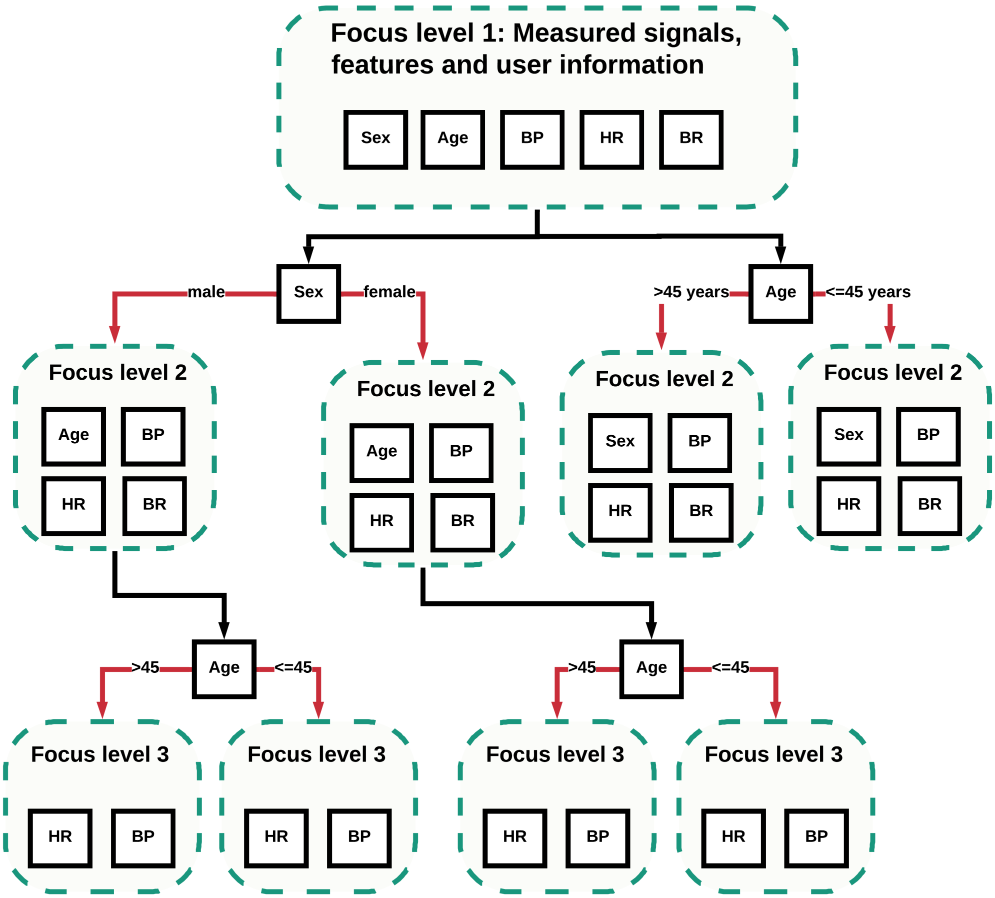
In Fig. 6b we provide a simple example of the decision-making tree with three focus levels. In Fig. 6b, three "Focus

TABLE 2. The notation used in this paper.

Notation	Definition
FA	The acceptable false alarm by the specified policy
$\boldsymbol{HD}$	$\boldsymbol{HD} = [h_1, h_2, \ldots, h_I]$ is related to health conditions we would like to diagnosis. h_i can map to number as follow: {1= healthy, 2=1st disease for diagnosing, 3= 2nd disease for diagnosing, …, the last number = unhealthy but have unknown disease}
h	Current health number condition
M	Maximum desired focus level
m	The current focus level
k_m	The total number of global features in focus level m
k_m	Current global feature number
F_m^k	The total number of tree branches for the feature k in-focus level m
f_m^k	The current tree branch in the feature k in-focus level m
T_{mkf}	The total PACs number for branch f, feature k, and focus level m
t_{mkf}	The current local PAC number for branch f, feature k, and focus level m
$\boldsymbol{O}_{mkf}$	The set of all available features for branch f, feature k, and focus level m
o_{mkf}	The current feature of available features for branch f, feature k, and focus level m
$\boldsymbol{U}_{mkf\|h}$	The set of all available users in the database for branch f, feature k and focus level m *for given* h
$u_{mkf\|h}$	The current user for branch f, feature k, and focus level m for given h
$\boldsymbol{BD}_m^k(o_m^k(f_m^k), \boldsymbol{U}_m^k(f_m^k))$	The set of values for feature o for all available users in branch f, feature k, and focus level m
$\Delta B_{o_m^k}$	Discretization step
$\hat{B}_m^k$	estimate B_m^k
$b_{min}^{m,k,f,o}$	The minimum boundary for the healthy range for feature o and in branch f, feature k and focus level m
$b_{max}^{m,k,f,o}$	The maximum boundary for the healthy range for fea$\hat{B}_m^k$ture o and in branch f, feature k and focus level m
$\boldsymbol{P}(\hat{\boldsymbol{B}}_m^k(o_m^k(f_m^k))\|\boldsymbol{HD})$	Bayesian generative model
$\boldsymbol{P}(\boldsymbol{HD}, f_m^k)$	Prevalence
$\boldsymbol{P}(\hat{\boldsymbol{B}}_m^k(o_m^k(f_m^k)))$	Evidence
$\boldsymbol{P}(\boldsymbol{HD}\|\hat{\boldsymbol{B}}_m^k(o_m^k(f_m^k)))$	Posterior
$r_{o_m^{kf}\|h}$	The current probability for the value of feature o for given h happen out of normal range
$\boldsymbol{r}_{o_m^{kf}\|h}$	The set of all probabilities for the value of a set of $\boldsymbol{O}$ for given h happen out of normal range
$\boldsymbol{C}_{mkf\|h}(FA=0)$	The set of all actions in the static library for zero FA policy
$c_{mkf\|h}(FA=0)$	The current action in the static library for zero FA policy
$umin$	The minimum required users for branch tree f and focus level m for reliable Bayesian model extraction.
$AF_{t_{mkf}}$	The related assurance factor for PAC number t_{mkf}
$rw_{t_{mkf}}$	The internal rewards as defined $(1 - AF_{t_{mkf}})$



(a)



(b)

FIGURE 6. **The CDS can create the proposed decision-making tree (a) the general schematic, (b) a simple example of the general schematic. BP: Blood Pressure, HR: Heart Rate, BR: Breathing Rate.**

levels" are shown. In "Focus level 1," the measured signals, User information, features, actions, and desired false alarm rate are the inputs. Then, the CDS check all five features (Sex, Age, BP, HR, and BR), and computes the internal rewards. If the internal reward is below a predefined threshold, the CDS increases the focus level.

In "Focus level 2", for both branches of "sex" and "age," the system arbitrarily selects one of them, for example, "sex." Then, it computes the internal rewards (either male or female) based on the other four features (Age, BP, HR, and BR) and compares it to a predefined threshold. If the threshold for "sex" is not met, then the system switches to "age" and similarly computes the internal rewards and compares it to a predefined threshold. If the predefined threshold is again not met, then the system switches back to the "sex" branch and increases the focus level to "Focus level 3". In this example, the CDS continues in this way subject to eqs. (2) and (3) given below. The proposed idea of the decision-making tree is inspired by the focus level concept in a human brain [1], the focus level concept in CDS [3], and the MDs decision-making approach.

To create the decision-making tree and implement the focus levels in CDS, we use Algorithm 1. This proposed algorithm updates the Users and features for each tree branch f (e.g., male or female), tree node k (e.g., sex), and focus level m (e.g., second focus level). Fig. 6b shows an example of the decision-making. These updates can be done by calculating the probability of the branch in the database, which can be extracted as $P_f^{(k_m m)}$. The extracted $P_f^{(k_m m)}$ is easily found using the probabilities extracted for the discretized node feature k (see Algorithm 1) in focus level $(m-1)$ with $m>1$.

The tree branches can be binary, because traditionally, some features have two values. For example, someone can be either a smoker or non-smoker, drinks alcohol or does not drink alcohol. However, for some features such as the age, more than two branches can be created if there are enough number of instances in the training database. Therefore, eq. (2) shows the condition that can be used for branching using the feature statistically discretized (FSD) approach.

$$\text{ceil}(\textit{length of } \boldsymbol{U}_{(m-1)}^{k_{(m-1)} f_{k_{(m-1)}}} \times \boldsymbol{P}_f^{(k_m m)}) \geq umin \tag{2}$$

$$umin = \frac{5}{Threshold}. \tag{3}$$

In eq. (2), $umin$ is the minimum required number of Users for branch tree f and focus level m for reliable model extraction. In eq. (3), the *Threshold* is a predefined acceptable diagnostic error by the system. This minimum value for the number of Users can be found using a Monte-Carlo approach for extracting reliable conditional probabilities or a Bayesian generative model. Then, the information related to this decision tree can be saved in the perceptor and executive memory.

Besides, if the *Threshold* (see eqs. (2) and (3)) for the internal reward is not meet for all branches of the global feature of k_m in focus level m (see Fig. 6a), then the perceptor will use the global feature of $(k+1)_m$ for branching. However, if at least one branch of k_m achieves the *Threshold*, then the executive increases the focus level m to focus level $(m+1)$.

In each focus level, the CDS learns which features can provide new information about the unhealthy conditions of Users. The main advantage of this method is that even without a database, when the health condition of the User is known as no disease, the system can extract the normal ranges from sensor(s) measurements. However, the information about important features (i.e., the features that provide information about a User's health) helps the CDS to find important features for doing purposeful and cognitive actions.

For example, a smart e-Health home may actuate the blood pressure, weight, and heart rate sensors to diagnose tachycardia. However, cognitive actions like those taken by MDs can only actuate the heart-rate sensor for the diagnosis. Also, more focus levels can reduce the natural diagnosis error by default. Here, the natural diagnosis error means that there is a wrong decision due to incorrect reasoning. A wrong decision does not include errors due to measurement noise or failures of the sensors.

As an example, it is known that the normal fractional carbon monoxide in human blood is less than one percent [44]. As a result, more than one percent of carbon monoxide in the blood may need an alarm and treatment actions related to carbon monoxide in the immediate environment. However, with more evidence, the CDS may find that the Users are smokers. Therefore, more focus levels or branches such as how many cigarettes are smoked per day, the interval between smoking cigarettes and cigarettes brand will be implemented.

The normal amount carbon monoxide in the blood of a healthy smoker may be higher than one percent. As an example, for a smoker who smokes one pack of cigarettes per day, their normal carbon monoxide level in the blood can reach 3% to 6%. This is increased to 6% to 10%, when smoking two packs per day, and increases to 20% for three packs a day [44]. Therefore, the adaptive training system in the CDS perceptor makes it flexible in diagnosing severe health issues. However, if we consider smoking or tobacco addiction as the disease, then the scenario can be different.

Algorithm 1 shows how this training part is working in our CDS. Algorithm 1 shows how the CDS can find the important features in any arbitrary focus level. It uses a vector line for the calculation to reduce the algorithm complexity as much as possible. Therefore, it can be trained and can make a decision dynamically for real-time situations in a smart e-Health home. In this way, CDS is a suitable platform for ADMS.

In Algorithm 1, in line 8, we assumed from focus level 3 that the redundant tree branches should be removed. For example, branching from sex in focus level 2 ($m = 2$) to age in focus level 3 ($m = 3$) (see Fig. 6b) is similar to branching from age in focus level 2 ($m = 2$) to sex focus level 3 ($m = 3$).

MODEL LIBRARY IN THE PERCEPTOR AND ACTIONS LIBRARY IN THE EXECUTIVE

Normally, a MD can diagnose someone with diseases even if they did not diagnose such diseases before. Therefore, a semi-MD's approach is used for smart e-Health home applications. Here, the ADMS layer should be flexible enough to work with different features, conditions and policies. In addition, the ADMS may face two challenges in a smart e-Health home.

1. There is no database, but we can extract and measure the features of current Users. The designed CDS need to be trained using a database, which is a collection of data corresponding to a data matrix table. In the database table, every column of the table is for a specific feature, and each row corresponds to a given User with measured data and known health status. We can extract the normal range for such a person with known health conditions and when they are healthy. Then,

Algorithm 1 Creating a Decision Tree and Updating Users and Features Database for Related Branches

Input: The observables and features from the database for each User for the focus level m, global feature k, branch tree f, policy, labeled dataset, actions space $\boldsymbol{U}_{(m-1)}^{k_{(m-1)}f_{k_{(m-1)}}}$, $\boldsymbol{O}_{(m-1)}^{k_{(m-1)}f_{k_{(m-1)}}}$
Output: For the new branch, focus level and global feature k, the new User set of $\boldsymbol{U}_m^{k_m f_{k_m}}$, the new features set of $\boldsymbol{O}_m^{k_m f_{k_m}}$
Initialization:

for $m = 2$ to M **do**
for k = set of $\boldsymbol{O}_{(m-1)}^{k,f}$ **do**
Statistical discretization for features can have more than two branches [Age, weight, height, heart rate, sleep, …], $f_{k_m m} \in \{1, 2, \ldots, F_{k_m m}\}$
Creating a binary branch for: [sex, smoking, drug, family disease history, drinking, night sleep, activity/rest, $\cdots$], $f_{k_m m} \in \{1, 2\}, F_{k_m m} = 2$

Updating Users in the database for the branch of each tree depend on the probability
for set of $f_{k_m m} \in \{1, 2, \ldots, F_{k_m m}\}$ **do**
Calculate the $P_f^{(k_m m)}$: use previous probabilities extracted in previous focus level
$\boldsymbol{U}_m^{k_m f} = P_f^{(k_m m)} \boldsymbol{U}_{(m-1)}^{k_{(m-1)}f_{k_{(m-1)}}}$
if $m > 2$ **then**
$\boldsymbol{O}_m^{k_m f_{k_m}} = \boldsymbol{O}_{(m-1)}^{k_{(m-1)}f_{k_{(m-1)}}} - \{1, \ldots, k\}$
End if
End for
End for
End for

the CDS can find if a User has some diseases by extracting some features that seem abnormal.

2. We have enough number of instances in the database that includes healthy and unhealthy Users. However, we may have a failure of one or more sensors. In this case, the CDS should be able to find alternative features (i.e., features of available sensor signals that can help to diagnose a User's health state) in related available actions to keep the performance reliable enough. Therefore, in a real-time situation, the CDS algorithms should take proper alternative actions quickly.

The proposed CDS for ADMS in this paper can tackle conditions 1 and 2 (see Section VII.A later), which are important for improving the system's reliability in a health-care scenario. To meet challenges 1 and 2, we should use an algorithm with low complexity. The first training level is to find the model and the normal ranges. For extracting the posterior, the following probabilities should be calculated, $\boldsymbol{P}(\hat{\boldsymbol{B}}_m^k(o_m^k(f_m^k))|\boldsymbol{HD})$ (*Bayesian generative model*), $\boldsymbol{P}(\boldsymbol{HD}, f_m^k)$ (*prevalence*) and $\boldsymbol{P}(\hat{\boldsymbol{B}}_m^k(o_m^k(f_m^k)))$ (*evidence*) The

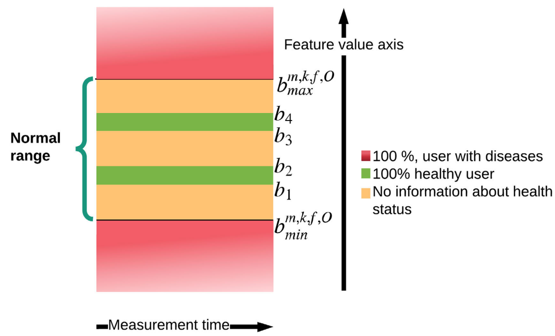


FIGURE 7. **The possible feature value and ranges for users with and without diseases in the database.**

$\boldsymbol{P}(\boldsymbol{HD}|\hat{\boldsymbol{B}}_m^k(o_m^k(f_m^k)))$ (*posterior*) can be calculated using the Bayesian equation as follows:

$$\boldsymbol{P}(\boldsymbol{HD}|\hat{\boldsymbol{B}}_m^k(o_m^k(f_m^k))) = \frac{\boldsymbol{P}\left(\hat{\boldsymbol{B}}_m^k(o_m^k(f_m^k)\right)|\boldsymbol{HD})\,\boldsymbol{P}\left(,f_m^k\right)}{\boldsymbol{P}(\hat{\boldsymbol{B}}_m^k(o_m^k\left(f_m^k\right)))}, \quad (4)$$

where the $\boldsymbol{HD} = [hd_1, hd_2, \ldots, hd_H]$ can correspond to $\boldsymbol{HD} = [1, 2, \ldots, H]$.$\hat{B}_m^k(o_m^k\left(f_m^k\right))$ is the discretized range for the feature number o, focus level m, main feature k and branch number f. In the database, the normal range means that 100% of the values of the features of Users without diseases fall in this range that can be calculated (see Fig. 7) using:

$$\sum_{\text{set of } \hat{\boldsymbol{B}}_m^k(f_m^k)} \boldsymbol{P}(h = 1|b_{min}^{k,m}(o_m^k(f_m^k)) \leq \hat{\boldsymbol{B}}_m^k(o_m^k(f_m^k)) \leq b_{max}^{k,m}(o_m^k(f_m^k))) = 1. \quad (5)$$

Algorithm 2 shows the model extraction by the perceptor for creating the model library. Also, Algorithm 2 creates the actions library by the executive. As a result, the actions can be assumed to be the same as MDs decision-making [23]. In addition, in common with MDs, these normal ranges can be dynamically changed by receiving new information or knowledge after dynamically updating the database. In addition, the normal ranges can be varied with more focus for better accuracy.

In the executive training mode, the executive extracts actions and their weights for low false alarm policy (diagnostic test). Algorithm 2 shows how these extractions are performed (See Fig. 7 also). If we have the features that provide information about Users with diseases (that the feature value is in the red region, see eq. (5) also), then this can be considered as an important feature. Data training is conducted using Algorithm 2. Then, Algorithm 3 can refine actions in the executive training mode. This means that the executive keeps important actions for diagnosing between healthy and unhealthy situations. Important actions mean that the features that can provide new information about an unhealthy situation. Algorithm 3 helps the system to use less memory, and because it has low

Algorithm 2 The Proposed System for the Training Mode of the Perceptor and Executive for Tree Branch of f, Node Feature k and m Focus Level

Input: The observables and features from the database for each User for the focus level m, node feature k, branch tree f, policy, database of Users and health conditions vector.
Output: Actions weight matrices $\boldsymbol{R}_{O_m^{kf}|\boldsymbol{HD}}$, actions matrices $\boldsymbol{C}^{kf}_{m|\boldsymbol{HD}}$, probabilities and the normal range for each feature
Initialization:
Calculate $\boldsymbol{U}^k_m(f^k_m)$ using $\boldsymbol{BD}^k_m$
Calculate the feature matrix number $\boldsymbol{O}^k_m(f^k_m)$

Perceptor training mode
for $o^k_m(f^k_m)$ = set of $\boldsymbol{O}^k_m(f^k_m)$ **do**
Updating corresponding features signals in the database
$\boldsymbol{B}^k_m = \boldsymbol{BD}^k_m(o^k_m\left(f^k_m\right), \boldsymbol{U}^k_m\left(f^k_m\right))$
The maximum and minimum value for the feature $o^k_m(f^k_m)$
Calculate $B^{km}_{min}(o^k_m(f^k_m))$ and $B^{km}_{max}(o^k_m(f^k_m))$
Discretization step
Calculate $\Delta B_{o^k_m}$
Estimation by discretization
Calculate $\hat{\boldsymbol{B}}^k_m\left(o^k_m\left(f^k_m\right)\right)$
for h = set of $\boldsymbol{HD}$ **do**
Calculate $\boldsymbol{P}(\hat{\boldsymbol{B}}^k_m(o^k_m\left(f^k_m\right))|h)$, $\boldsymbol{P}(h, f^k_m)$, $\boldsymbol{P}(\hat{\boldsymbol{B}}^k_m(o^k_m\left(f^k_m\right)))$ and $\boldsymbol{P}(h|\hat{\boldsymbol{B}}^k_m(o^k_m\left(f^k_m\right)))$
Save probabilities in the perceptor library
End for
Normal ranges calculation
Calculate $b^{km}_{min}(f^k_m, o^k_m(f^k_m))$ and $b^{km}_{max}(f^k_m, o^k_m(f^k_m))$
Maximum cell number in normal ranges
Calculate $N^{kf}_m = \frac{b^{kmf}_{max} - b^{kmf}_{min}}{\Delta B_{o^{kf}_m}}$
Save normal ranges and N^{kf}_m in healthy ranges in the perceptor library

Executive training mode for FA = 0
$r_{o^{kf}_m|h=1} = 0$
for $h = 2$ to **H do**
$r_{o^{kf}_m|h} = 0$
Build actions library for FA = 0
if $(\boldsymbol{P}(\hat{\boldsymbol{B}}^k_m(o^k_m\left(f^k_m\right)) < b^{km}_{min}\left(f^k_m\right)|h) > 0$ or $\boldsymbol{P}(\hat{\boldsymbol{B}}^k_m(o^k_m\left(f^k_m\right)) > b^{km}_{max}\left(f^k_m\right)|h) > 0)$ and $(\boldsymbol{P}(h, f^k_m), > 0)$ **then**
$r_{o^k_m|h} = (\boldsymbol{P}\left(\hat{\boldsymbol{B}}^k_m(o^k_m\left(f^k_m\right)\right) < b^{km}_{min}\left(f^k_m\right)|h) + \boldsymbol{P}\left(\hat{\boldsymbol{B}}^k_m(o^k_m\left(f^k_m\right)\right) > b^{km}_{max}\left(f^k_m\right))|h)$
Updating the action related to the current signal feature
$c^k_{mh|o^k_m}\left(f^k_m, FA = 0\right) \leftarrow sensorforo^k_m\left(f^k_m\right)$
End if
End for
End for

Algorithm 3 Actions Refining, f^k_m *and* $FA = 0$ and Just Checking If Someone Is Healthy Or Not, Search for, If There Is New Information About Any Users

Input: Observables and features from the database for each User for the focus level m, node feature k, branch tree f, policy, training database, actions space $\boldsymbol{C}^{kf}_{m|\boldsymbol{HD}}$, $\boldsymbol{BD}^{kf}_m$, $\boldsymbol{R}^{kf}_{m|\boldsymbol{HD}}$
Output: The important actions of $\boldsymbol{C}^{kf}_{m|\boldsymbol{HD}}$

Initialization:
buffer $= 0$, $\boldsymbol{newinf} = \emptyset$

for $h = 2$ to **H do**
$r\_buf = r_{\boldsymbol{O}^{kf}_m|h}$
Sort the r_buf in descending order and remove 0 value elements
for $o^k_m(f^k_m)$ = set $\boldsymbol{O}^k_m(f^k_m)$ **do**
for u = set of $\boldsymbol{U}^k_m(f^k_m)$ **do**
if $BD^k_m(o^k_m\left(f^k_m\right), u) > b^{km}_{max}(f^k_m, o^k_m(f^k_m))$ **or** $BD^k_m\left(o^k_m\left(f^k_m\right), u\right) < b^{km}_{min}(f^k_m, o^k_m(f^k_m))$ **then**
$newinf_u = 1$
end if
end for
s = summation of $\boldsymbol{newinf}$
if summation of $s <=$ Buffer **then**
$c^k_{mh|o^k_m}\left(f^k_m, FA = 0\right) \leftarrow \{\}$
$r_{o^{kf}_m|h} = 0$
end if
Buffer $= s$
end for
end for

complexity, it can reduce additional sensors usage and activation. This results in energy saving and longer sensors' lifetimes.

The CDS stores the features of important actions in the executive actions library. The action space can be defined as follows:

$$C = \left\{ c_k \;\middle|\; c_k = \begin{array}{l} Actuate\ sensor\ c^k_{(mh|o^k_m)}(f^k_m, FA = 0) \\ \quad for\ o^k_m(f^k_m) \\ Internal\ commands = Increase\ m, \\ \quad m < M \end{array} \right\}. \tag{6}$$

Here, c_k is the action in PAC number k and m is the focus level (see Table 2). The action space means that all possible actions can be done during a User's health prediction mode by the executive.

D. PREDICTION MODE: PERCEPTOR AND EXECUTIVE

Algorithm 4 shows the implementation of the CDS prediction mode. Also, this algorithm shows how the CDS performs when we have a User in a smart e-Health home with an unknown health situation. In this condition, the system randomly chooses an action from the actions library. Then after applying new actions, the assurance factor (AF) is calculated using eq. (7):

$$AF_{t_{mkf}} = \frac{\boldsymbol{P}(h, f_m^k) r_{j|h}}{\boldsymbol{P}(h > 1, f_m^k)} + AF_{(t_{mkf}-1)}. \tag{7}$$

The assurance factor measures the expected assurance about the decision (see section VI.A also) after doing current action $c^k_{(mh|o^k_m)}(f^k_m, FA = 0)$. Therefore, the internal reward can be calculated as follows:

$$rw_{t_{mkf}} = (\mathbf{1} - AF_{t_{mkf}}). \tag{8}$$

Then, the system, depending on the health conditions related to $hd_2, \ldots, hd_H$ (see eq. (1)), will apply related actions. Related actions mean asking for related information to the User or activating sensors. For each disease, the executive activates the sensors to obtain maximum information about the health conditions hd_h and $h > 1$ (see eq. (1)) using the planning and learning sections (see Algorithm 4). Reinforcement learning will be done once, when the database is updated with new information.

E. COMPLEXITY OF THE PROPOSED ALGORITHMS

Typical machine learning (ML) techniques select the key features based on different methods such as principal component analysis. Then, the typical ML techniques use all key features simultaneously to extract the model. After this, the extracted model by ML can be used for decision making in a multi-dimensional space. In this approach with typical ML techniques, modeling, and decision making for Users with unknown health conditions will take much time. However, in our proposed algorithms, the key features can be found in the executive training mode (See Algorithms 2 and 3).

In addition, for decision making, the CDS will check all features one by one in each PAC. Therefore, instead of creating a multi-dimensional feature space, the CDS checks the features sequentially. Because of checking the features one by one, CDS can decide in a one-dimensional space. Then, the decision-making accuracy is improved using the PAC concept. As a result, the proposed algorithms are fast enough for real-time healthcare applications such as in a smart e-Health home.

Having fast enough algorithms for real-time healthcare applications are an important advantage of our proposed approach using CDS. However, it should be noted that the required memory for saving models and action space will increase as the number of focus levels are increased in the CDS. In addition, to address increasing algorithms complexity by increasing the number of focus levels, we can define a bound for the maximum possible focus level as the *Complexity threshold*. Then, we can calculate the total acceptable branches of the decision tree as:

$$F_m^{total,k} = \sum_{m=1}^{M} F_{k_m},$$
$$\text{and, } F_m^{total,k} \leq \textit{Complexity threshold} \tag{9}$$

Algorithm 4 CDS User-Health Prediction (Planner, Reinforcement Learning and Policy in Executive and Running Diagnostic Test in the Perceptor)

Input: The observables and features from the database for each User for the focus level m, node feature k, branch tree f, policy (FA=0), a database of Users, $\boldsymbol{HD}$ health condition vectors.

Output: Actions weight matrices $\boldsymbol{R}_{\boldsymbol{O}_m^{kf}|\boldsymbol{HD}}$, actions matrices $\boldsymbol{C}_m^{kf}$, probabilities and the normal range for each feature

Initialization:

$c_0 \leftarrow$The actions, $[Vitalsignsandportablesensors, \cdots]$ apply on smart e-Health home User

Start advanced actions such as 12 leads ECG or ... if c_0 shows something abnormal or voices show pain, need help, or User saying directly, "I am not feeling well."

Load the normal ranges $\boldsymbol{b}_{min}^{m=1}(\boldsymbol{O}_{m=1})$, $\boldsymbol{b}_{min}^{m=1}(\boldsymbol{O}_{m=1})$, Actions weights $\boldsymbol{R}_{\boldsymbol{O}_{m=1}|\boldsymbol{HD}}$, Actions $\boldsymbol{C}^{\boldsymbol{O}_{m=1}}_{m=1|\boldsymbol{HD}}(FA = 0)$

$m = 1$ (focus level 1), $\boldsymbol{k}_{\boldsymbol{m=1}} = \emptyset, \boldsymbol{f}_{\boldsymbol{m=1}} = \emptyset, \boldsymbol{t} = \emptyset$, Decision $= 1$, $Threshold =$ Acceptable estimated error.

$c^{o_{m=1}}_{m=1|h}(FA = 0) \leftarrow$ an action randomly selected from $\boldsymbol{C}^{\boldsymbol{O}_{m=1}}_{m=1|\boldsymbol{h>1}}(FA = 0)$

Apply to $c^{o_{m=1}}_{m=1|h}(FA = 0)$ the User $(h \in \{2, \ldots, H\})$

Extract features $o^{0,0}_{m=1}$ and $\boldsymbol{BD}^k_m(o^{k,f}_m, u_{test})$ and choose one of them with maximum $r_{o_{m=1}|h}$

Calculate $AF_{t_{m=1,00}} = \frac{\boldsymbol{P}(h,\mathrm{m}=1) r_{o_{m=1}|h}}{\boldsymbol{P}(h>1,\mathrm{m}=1)}$

calculate first internal rewards $as 1 - AF_{t_{m=1,00}}$,

for $k =$ set of $\boldsymbol{k_m}$ **then**
  for $f =$set of $\boldsymbol{f}_{\boldsymbol{mk}}$ **do**
    $t_{mkf} = 0$
  for $h = 2$ to **H do**
    $\boldsymbol{r\_buf} = \boldsymbol{r}_{\boldsymbol{O}_m^{kf}|h}$
  Sort the $\boldsymbol{r\_buf}$ decently and remove 0 value elements
    $\boldsymbol{O}_m^{kf}$ =update features based on elements on $\boldsymbol{r_{buf}}$

Planning

$\boldsymbol{C\_buf} \leftarrow \boldsymbol{C}^{mkf}_{h|\boldsymbol{O}_{mkf}}\ (FA = 0)$
$\boldsymbol{A}_{t_{mkf}} \leftarrow \boldsymbol{C\_buf}$

Reinforcement learning

for all actions $\left(c \in \boldsymbol{A}_{t_{mkf}}\right)$ **do**
  for $(j \in setof\, \boldsymbol{O}_m^{kf})$ *and* $(j \in$ *features extracted from* $c)$ **do**
    $AF^{cj}_{(t_{mkf}+j)} = \frac{\boldsymbol{P}(h, f_m^k) r_{j|h}}{\boldsymbol{P}(h>1, f_m^k)} + AF^{cj}_{(t_{mkf}+j)}$
    $rw^{cj}_{(t_{mkf}+j)} = 1 - (AF^{cj}_{(t_{mkf}+j)} + AF_{t_{mkf}})$
  End for
End for
$J \leftarrow \underset{\boldsymbol{J} \in \boldsymbol{A}_{t_{mkf}}}{\mathbf{argmin}} \left\{ \sum_j rw^{\boldsymbol{A}_{t_{mkf}} j}_{(t_{mkf}+j)} \right\}$
remove J from $\boldsymbol{C\_buf}$
Apply action (sensor activation) J on User

Algorithm 4 *(Continued).* CDS User-Health Prediction (Planner, Reinforcement Learning and Policy in Executive and Running Diagnostic Test in the Perceptor)

Run diagnostic test
20: $\boldsymbol{O_{buf}} \leftarrow$ *features of J*
21: $\boldsymbol{V_{O_{buf}}} \leftarrow$ ***save values related to*** $\boldsymbol{O_{buf}}$
30: **for** $(j \in setof\, \boldsymbol{O_{buf}})and(j \in setof\, \boldsymbol{O_m^{kf}})$ **then**
22: $t_{mkf} = t_{mkf} + 1$
Internal rewards calculation
23: $AF_{t_{mkf}} = \frac{P(h, f_m^k) r_{j|h}}{P(h>1, f_m^k)} + AF_{(t_{mkf}-1)}$
24: $rw_{t_{mkf}} = 1 - AF_{t_{mkf}}$
25: **if** $\boldsymbol{V}_j < b_{min}^{mkfj}$ or $\boldsymbol{V}_j > b_{max}^{kmfj}$ **then**
26: Alarm and disease diagnosis process
27: Decision = Unhealthy
28: **Return** Decision
29: **End if**
30: **End for**
31: **if** $\boldsymbol{C\_buf} \neq \emptyset$ **then**
32: Go to line 18
33: **End if**
34: **End for**
35: **if** $rw_{t_{mkf}}$ <= *threshold* **then**
36: Healthy life recommendations
37: Decision = Healthy
38: **Return** Decision
39: **elseif** *Users in focus level* $m+1 >= \underline{u_{min}}$ **then**
40: increase focus level
41: **else**
42: Run Screening process
43: **End if**
44: **End for**
45: **End for**

Furthermore, F_{k_m} corresponds to the maximum number of decision-making tree branches in focus level m and global feature k. For the desired predefined *Complexity threshold,* the CDS cannot increase the focus level more than M. However, in practice, the bound is determined by the lower value from either eq. (9) or eqs. (2) and (3), and that will be the maximum focus level.

VII. CASE STUDY: DIAGNOSING BETWEEN A USER WITH OR WITHOUT ARRHYTHMIA

Every year, about 326,000 out-of-hospital sudden cardiac arrest (OHSCA) can happen in the USA [45]. The median age of these persons is about 65 years. Unfortunately, the survival rate is 10-11%. It is important to know that the survival rate can improve to 33.3% if OHSCA happens in front of another person(s) [46]. More than 80% of sudden cardiac death is due to ventricular Arrhythmia [46], [47]. Globally, the prevalence of Arrhythmia is unknown. However, it is assumed that millions of people in the world have Arrhythmia [48]. Also, Arrhythmia is most common in persons older than 35 years [45], [48].

In the USA, 850,000 persons are hospitalized each year due to Arrhythmia [45]. Some researches show that at least 16-17% of Canadians are not aware that they have Arrhythmia disease [49]. Diagnosis of Arrhythmia is made by the 12-leads ECG [50] (see Fig. 8) or by using the Holter monitor for two or three weeks monitoring [51]. However, these methods can only diagnose about 50% of Arrhythmia [52]. Figures 8a and 8b show an example of 12-leads ECG output signal and ECG electrodes locations for leads on the human body, respectively.

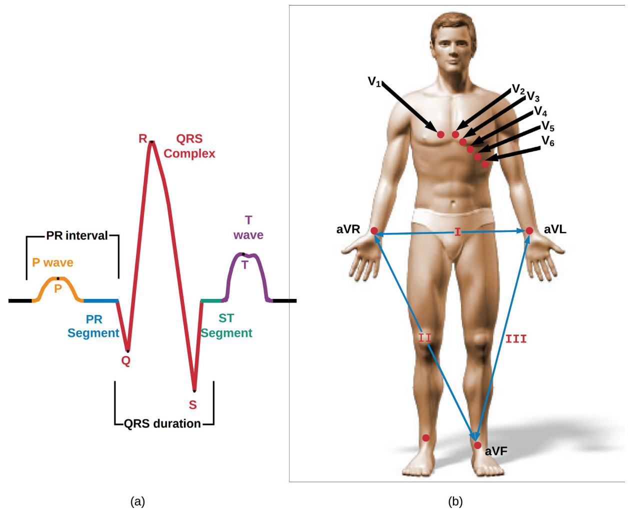


FIGURE 8. The 12 leads ECG (a) an example of an ECG signal (b) electrode places on the human body.

In the standard 12-leads ECG, six of the leads are known as "limb leads" because they are placed on the wrists and feet of the User. These six limb leads are known as leads I, II, III, aVF, aVR, and aVL (See. Fig. 8b). The letter "a" represents for "augmented," as aVF, aVR and aVL leads are calculated as a combination of leads I, II, and III. Therefore, as shown in Fig. 8b, aVF, aVR and aVL are the names for ECG leads for foot and right and left wrists, respectively. The leads V1-V6 are known as "precordial leads" because leads V1-V6 are placed on the chest (precordium). Also, in this paper, leads V1-V6 corresponds to the ECG leads 1-6.

In this section, simulation results are presented for the Arrhythmia case study as a proof-of-concept study, using the proposed algorithms in the previous sections. We used the MATLAB Arrhythmia database [53] for the case study simulations. Also, the MATLAB database is known as the UCI Machine Learning Repository [54]. The MATLAB Arrhythmia database includes 279 extracted features including age, sex, height, weight, heart rate and 12-leads ECG features (see Fig. 8) such as T wave angle, T wave duration and QRS angle for each lead for 452 persons (for ECG signal feature definition see [55]).

The MATLAB Arrhythmia database is mentioned as one of the hardest databases for classification [56]. This is because the MATLAB Arrhythmia database has a few numbers of instances (i.e., 452 instances) and there are some missing features such as J angle or heart rate (key feature) for some Users. Therefore, these missing features can simulate sensor failure in practical applications that were mentioned in Section VI.C.2.

TABLE 3. Prevalence in MATLAB database (prior).

Class	Users out of 452 persons	Prevalence %
1 (Healthy)	245	54.20
2 (Arrhythmia)	44	9.73
3 (Arrhythmia)	15	3.32
4 (Arrhythmia)	15	3.32
5 (Tachycardia)	13	2.88
6 (Bradycardia)	25	5.53
7 (Arrhythmia)	3	0.66
8 (Arrhythmia)	2	0.44
9 (Arrhythmia)	9	1.99
10 (Arrhythmia)	50	10.11
11 (Arrhythmia)	0	0
12 (Arrhythmia)	0	0
13 (Arrhythmia)	0	0
14 (Arrhythmia)	4	0.89
15 (Arrhythmia)	5	1.11
16 (unknown, unhealthy)	22	4.87
Total number of persons with Arrhythmia	207	45.80
Female	249	55.09
Male	203	44.91
Age >45 years	242	53.54

The persons in the MATLAB database were classified into 16 classes. Class 1 is for no Arrhythmia, classes 2-15 correspond to different classes of Arrhythmia and class 16 corresponds to an unknown Arrhythmia class. Also, the prevalence and demographics of Users in the MATLAB Arrhythmia database are presented in Table 3. Arrhythmia is defined as irregular, too fast (tachycardia) or too slow (bradycardia) heartbeats.

A. SIMULATION PARAMETERS AND RESULTS

Here, we use the leave-one-out cross-validation (LOOCV) for training and verifying the performance of the CDS. In this method, all Users except one are used for training, and the excluded User is used for testing and health-state diagnosis accuracy. This process is repeated for N = 452 times for N Users based on the MATLAB Arrhythmia database. The advantage of LOOCV is that the entire database can be used for training and testing. The error rate can be calculated as the average of the error rate of each iteration. More detailed information related to the LOOCV validation technique can be found in [57].

The health *space* is $hd_h \in \{hd_1 = healthy, hd_2 = AC2, hd_3 = Arrythmia\ class\ (AC)\ 3, hd_4 = AC4, hd_5 = AC5, hd_6 = AC6, hd_7 = AC7, hd_8 = AC8, hd_9 = AC9, hd_{10} = AC10, hd_{11} = AC14, hd_{12} = AC15, hd_{H=13} = Others(class\ 16\ of\ database)\}$ where $h \in \{1, 2, \ldots, H = 13\}$. Based on the proposed algorithms, examples of some actions for Focus level 1, which can provide information about the Users' health status, are shown in Figs. 9(a)-(h) for the most prevalent Arrhythmia classes (see Table 3) in the MATLAB database. Also, Table 4 shows the list of examples of actions in Fig. 9 for the Focus level 1. Here, the CDS can diagnose Arrhythmia (see Fig. 9) by activating the specific electrodes in the 12-leads ECG (see Algorithm 4 and section VI.B).

TABLE 4. List of most prevalent arrhythmia classes and example of actions in figure 9 (focus level 1).

Class	Action example to diagnose this class (Sensor) (see Fig. 8b)	Feature for this action (see Figs. 8a and 8b)
2 (Arrhythmia)	Lead 6 (V6)	T wave amplitude
3 (Arrhythmia)	Lead 2 (V2)	Q wave duration
4 (Arrhythmia)	Lead 2 (V2)	DII N intrinsic (or intrinsicoid) of deflection [53]
5 (Tachycardia)	Heart rate	Heart rate per minute
6 (Bradycardia)	Heart rate	Heart rate per minute
9 (Arrhythmia)	Lead 2 (V2)	QRS amplitude
10 (Arrhythmia)	Lead 1 (V1)	QRS amplitude
16 (Unknown, Unhealthy)	aVF	P wave amplitude

Actuating only required electrodes results in lower sensors usage, lower energy usage and longer sensor lifetime. We set the *Threshold* for the estimated error at 2.5% (i.e., (1-*AF*)≤0.025 and 97.5% assurance factor (*AF*), see eq. (3), and section VI.C.1). Therefore, using eq. (3), the *umin* is equal to 200. Due to the lack of enough Users in the MATLAB database, the CDS maximum focus level is 2 with two tree branches (see eq. (2), and *umin*=200).

In Focus level 2, CDS uses the sex feature in the MATLAB database only to extract the important features that depend on smart e-Health home User. Most features in the Arrhythmia database have a non-Gaussian distribution. For example, Fig. 10 shows the conditional probability for the heart rate (HR) without and with Arrhythmia, which has a non-Gaussian distribution.

As mentioned before, we used the LOOCV method to verify the proposed CDS algorithms performance for diagnosing the Arrhythmia as a proof-of-concept example. Therefore, the User-under-test is removed from the Arrhythmia database. Then, at Focus level 1, using Algorithm 2 in the CDS training mode, the perceptor creates the model library, and the executive extracts the important features. Next, in Algorithm 3, these important features are refined and the features that did not provide new information are removed. Also, in Algorithm 3, an action space for the executive is created.

Using Algorithm 4 (also, see Section VI.B) at Focus level 1, the executive (see Algorithm 4 and Section VI.B, planner part) apply the actions from the actions library for diagnosing the disease class of $hd_{h>1}$. If the User is healthy

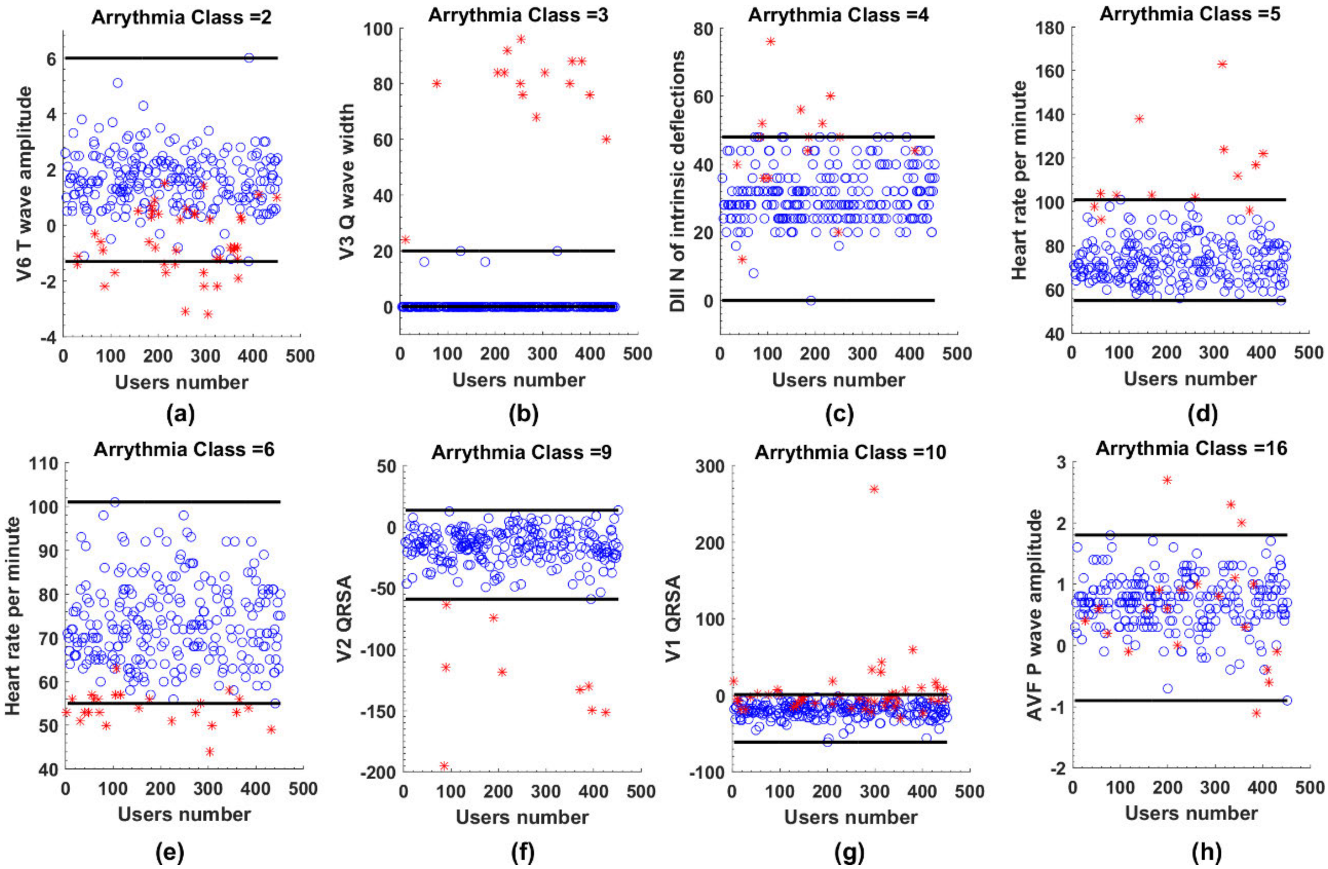


FIGURE 9. Examples of the actions selected by the CDS for the most prevalent class of Arrhythmia in the MATLAB database (Focus level 1), blue open circles are a healthy person and Red stars are users with Arrhythmia. Normal ranges are between the solid black lines. V1: ECG lead 1, V2: ECG Lead 2, DII N intrinsic (or intrinsicoid) of deflection: Deflection of ECG lead II (see. Fig. 8 also).

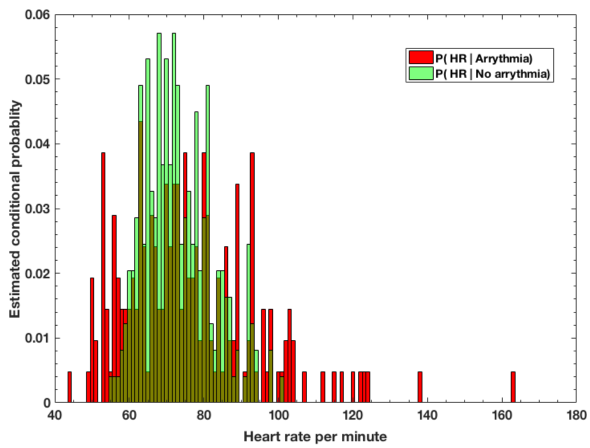


FIGURE 10. The conditional probability of heart rate for given health conditions: Light transparent green bar is persons without Arrhythmia, and the red bar is for persons with Arrhythmia (Focus level 1).

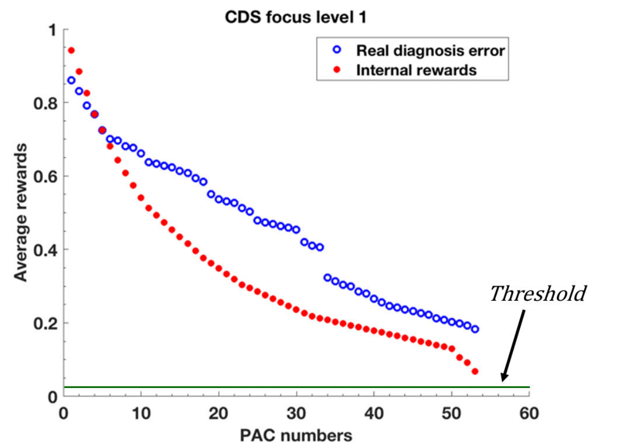


FIGURE 11. Internal rewards vs. real diagnosis error, in focus level 1 using the LOOCV approach.

and (1-AF)>$Threshold$ = 0.025, then the CDS checks the User's health for $hd_{h+1\leq H=13}$. Here, in Fig. 11, the average results are shown for the estimated diagnosis error as filled circles (i.e., 1-AF) and the average real error as the open circles for the LOOCV method. Also, in Fig. 11, the average diagnosis of the real error is 18% at PAC number 52 (the CDS does not know the real error shown as the open circles). However, at PAC number 52, the assurance factor is ∼95%, and the estimated error of 0.05 is larger than the $Threshold = 0.025$.

As the focus level can be increased (see Algorithm 1) for branches male and female (see eq.2, Table 3: male data:

TABLE 5. Summary of proposed CDS performance for the proof concept case study on arrythmia.

CDS Global actions	False Alarm (1-Specificity) %	Diagnosis error (1-Sensitivity) %	Total error %	If Healthy	If Arrhythmia
Focus level 1	0	18	8.3	Next global action	Alarm, treatment actions
Focus level 2 (Male or female)	0	10	4.6	Next global action	Alarm, treatment actions

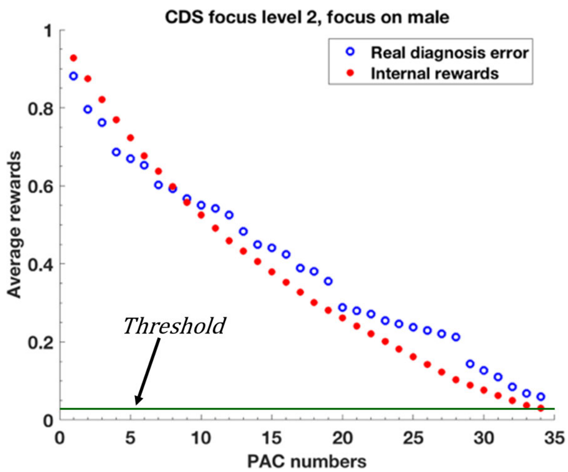
203 and female data: 249 > $umin$=200), the executive sends internal commands to the perceptor to increase the focus level. The same training procedure is done for creating models and actions library in Focus level 2 for the perceptor and the executive, respectively (see Algorithms 2 and 3, and section VI.B).

The CDS rechecks the User, depending on whether the User is male or female (see Algorithm 4 and Section VI.B: Prediction part). Fig. 12a shows that the real final diagnosis error for the male Users is 5% at PAC number 34, and the estimated error is 0.02, which is less than 0.025, the predefined threshold. So, if all the measured features of a male User are normal, then the system can claim him as a healthy User and can send him to receive healthy living and disease prevention recommendations.

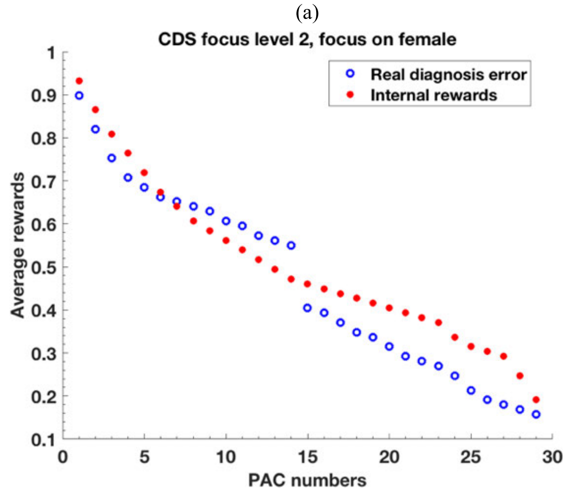
Fig. 12b shows that the real diagnosis error (open circles) for female Users is 14%. Also, the estimated error for female Users using the assurance factor is ~0.18 (filled circles), which is higher than 0.025 as the predefined threshold. Also, due to the lack of the number of available instances in the Arrhythmia database, the focus level cannot increase to 3. Therefore, for the female Users, if the CDS do not find any disease class in the female User signals, then the female Users are sent for the screening process.

In Table 5, the total real diagnosis accuracy improved from 82% to 90% by increasing Focus level 1 to Focus level 2 for the low false alarm policy (FA=0). In addition, the total accuracy improved from 91.7% in Focus level 1 to 95.4% in Focus level 2. Another important improvement is the reduction in required number of PAC. The reduction of the maximum required PAC numbers can be seen by comparing Fig. 11 and Figs 12a and 12b. In Focus level 1 (if the system does not know about the User's sex), we needed 53 PACs. However, in Focus level 2 for males, number of PACs is reduced from 53 to 34. Similarly, the required PACs for female Users are reduced to 29. Thus, the usage of the sensors can be reduced in Focus level 2.

In Figs. 11 and 12, there are some breaks in the real error graphs (open circles). The actual error is unknown in the proposed CDS. In addition, the estimated error (filled circles) using the internal reward (1-AF) is an approximation of real diagnosis error that cannot follow these breaks in the real diagnosis error (open circles). By seeing the description of Algorithm 4 and the high-level algorithm presentation in Section VI.B, the executive does the actions to check if the User has a certain disease class, for example, the second disease class of Arrhythmia (hd_2). If the User does not have the disease class hd_2, then the CDS will check if the User



(a)



(b)

FIGURE 12. Focus level 2, internal rewards vs. real diagnosis error, using the LOOCV approach (a) male, (b) female.

has the disease hd_3 with other actions in action space for hd_3. This process continues until the disease class hd_H (here, for this Arrhythmia database, it is $hd_{H=13}$). For example, the first best action that is selected for diagnosing the disease class of hd_3 may perform better than the last action for diagnosing the disease class hd_2. Therefore, the breaks in the real diagnosis error are due to switching for diagnosing different Arrhythmia classes.

B. PROPOSED ALGORITHMS COMPLEXITY AND RUN TIME

It is useful to compare the complexity of the proposed method in this paper with the techniques presented in Table 1.

We run all algorithms on a Microsoft Surface-Pro with Intel ®Core TMi706650U CPU @ 2.20GHz 2.21 GHz, 16 GB RAM, system type 64-bit Operating System x64-Based processor using MATLAB. For the proposed algorithms in Table 1, the runtimes are between 2 to 10 minutes. However, this runtime for the proposed CDS is about 35.8 seconds (including LOOCV for 452 times modeling and decision making for two focus levels). Therefore, the proposed algorithms using the CDS concept are much faster than the typical methods such as support vector machine [9]. As a result, the proposed algorithms need less than 80 ms to do real-time modeling in two focus levels and to make a decision about a User in a smart e-Health home. Besides, the proposed algorithms are fast enough for real-time applications in a smart e-Health home.

VIII. RESEARCH CHALLENGES AND FUTURE WORK

In this paper, we implemented the first step of the ADMS for "*Diagnostic test* (*low false alarm policy*) *for someone who has a disease or no disease.*" However, in future, the algorithms should be extended for other ADMS policies, step by step to include: "*Screening applications*", "*Diagnosing disease class, i.e., someone has known diseases, but ADMS should diagnose the disease class*", "*Search for and treat the cause of disease*", "*Healthy life recommendations for illness prevention*", and "*Tracking, recovery, and healing after surgery or medical treatment*".

A. ADMS EXTENSION FOR HEALTHCARE APPLICATIONS

For extending ADMS to other healthcare policies using the CDS, proper datasets are necessary for training the CDS and preventing blind trial and error by the executive reinforcement learning (RL). In addition, for *prevention policy* and *tracking recovery and healing after surgery or medical treatment*, the fundamental definition of prevention or tracking algorithm should be converted to an engineering description for implementation.

In addition, in prevention, the CDS should be able to predict the User's future health condition as the current CDS is designed to find health conditions at the present time. With predictive capabilities, the CDS can recommend to the User the required actions for the present time and help the User to maintain their good health condition. In addition, for the disease class diagnosis policy, the important question of: "*How many disease classes can be diagnosed by the CDS in the presence of unforeseen noise, disturbance, and false information?*" should be answered.

Therefore, for reliable diagnoses and reduced risks, the CDS should able to calculate the maximum number of possible disease classes that can be diagnosed at the present time. In addition, another open question to be studied is "*How the CDS can use a reduced dataset* (*e.g., some important sensors failed or are not functioning properly*) *with the ADMS and its policies to still make an accurate diagnosis?*" A preliminary form of this question is addressed in this manuscript, but more study is needed. Also, in future, we plan to extend

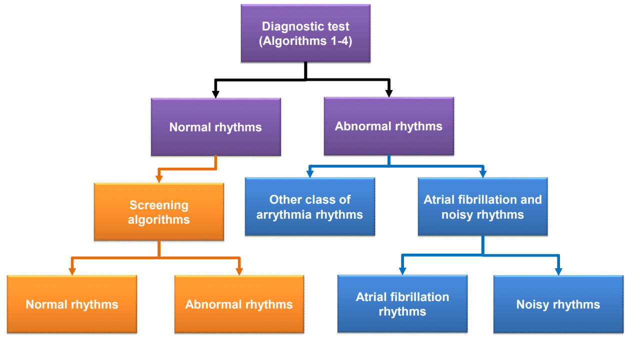


FIGURE 13. The schematic of extension of CDS based ADMS as the future work using CINC 2017.

the ADMS based on the CDS for other healthcare policies such as the screening process by extending the algorithms proposed in this paper. Finally, when these questions are answered, it will be possible to apply the CDS to health screening and disease prevention.

B. SECOND CASE STUDY ON ARRHYTHMIA RESULTS

Our next step is the extension of proposed algorithms in this paper for screening and disease class diagnosis applications. Therefore, we select the database of the PhysioNet Computing in Cardiology Challenge (CINC) 2017 [58] for the extension of our proposed algorithms.

The provided ECG recordings in the CinC 2017 database are collected using the FDA approved AliveCor KardiaMobile device. The KardiaMobile device is a commercial smart ECG sensor. KardiaMobile is a single-lead ECG system for personal use [58]. The database contains 8,528 single-lead ECG recordings from 9 s to just over 60 s. The sampling frequency of the AliveCore device for ECG recordings are 300 Hz. The database has ECG waveforms of normal rhythms (Healthy), atrial fibrillation (AF) rhythms, other rhythms (Other class of Arrhythmias) and noisy recordings.

In this paper, we have proposed diagnostic test Algorithms for training, refining and updating the CDS and how to use it for a User health diagnosis. In Fig. 13, we show a schematic representation of our future work on the CINC challenge dataset to extend the proposed CDS for screening and disease class diagnosis applications. For feature extraction, we used available MATLAB codes on the Physionet website [59]. The codes can extract up to 188 features from the ECG signals [59]. Then, we run the proposed CDS similarly to the first case study using the LOOCV validation. As a result, we provide the accuracy results in comparison to reported accuracy [58], [60]–[64] for this database in the published papers in Table 6.

Then we applied our proposed CDS for the diagnostic test on this database. The CDS meets the threshold 0.025 at Focus level 2 ($m = 2$) and global feature of 116 ($K_2 = 116$). Also, the tree branches for each global feature at Focus level 2 are three ($F_2 = 3$).

TABLE 6. Comparison between proposed work and related work published in literature for CINC 2017 database.

References:	Normal rhythms accuracy (Specificity) (%)	Abnormal rhythms accuracy (Sensitivity) (%)	F1-score (%)
[60], rank 1 [58]	-	-	83
[61], rank 1 [58]	-	-	83
[62], rank 1 [58]	-	-	83
[63], rank 5 [58]	89.9	71.4	82
[64], rank 5 [58]	-	-	82
Rule + Bagged tree	91	76	81
CDS (Focus level 2, $K_2 = 116$, F_2=3)	98.7	84.9	91.3

IX. CONCLUSION

In recent years, efforts were made to develop an autonomic computing layer in a smart e-Health home. However, currently, there are no comprehensive autonomic decision-making system (ADMS) or standards for the smart e-Health home autonomic computing layer. Practically, the ADMS functions needed are for real-time dynamic training or decision-making, screening, treatment, healing tracking as well as healthy living recommendations. In this paper, we proposed the first step for an ADMS based on a cognitive dynamic system (CDS) for running the diagnostic test and health situation understanding with a low false alarm policy.

The system architecture and algorithms are presented for health situations (i.e., healthy or unhealthy) diagnosis with low false alarm policy. To illustrate the application of our proposed system, a proof-of-concept case study is done on the Arrhythmia database. Our system provided an acceptable total accuracy of 95.4% that was achieved by increasing the focus level of the CDS. Also, for the proposed Arrhythmia case study, the run time of our algorithms is less than 80 ms for a User of the smart e-Health home. This time included time for the real-time modeling in two focus levels and for decision making.

The smart e-Health home can be implemented using IoT technology as a more generalized form of a cyber-physical system. In this scenario, the smart e-Health home using IoT technology must consider issues related to fog and edge computing and latency. Furthermore, the proposed, low complexity fast algorithms for the autonomic computer layer do not intensify these IoT technology issues because in the example provided, less than 80 ms was required time for training and prediction.

For the Arrhythmia case study as a proof-of-concept example, we used the database in which some key features such as heart rate were missing. It should be noted that missing key features can also be regarded as sensor(s) failure. Therefore, we could simulate the proposed CDS flexibility and reliability in the presence of the sensor(s) failure. Further, the 95.4% accuracy shows that the proposed CDS can find alternative actions for the relevant diagnostic tests.

In summary, for implementing the CDS, the following concepts are implemented in this paper: decision-making tree, inspiration from medical doctors (MDs) decision-making approach, converting data in the database to knowledge, prediction using the Bayesian model, and the characteristics of non-Gaussian and non-linear health features. These concepts are used in the presented CDS algorithms. The CDS can check one feature in each perception-action cycle (PAC). Checking one feature in each cycle makes the proposed algorithms simple and fast. Therefore, the presented algorithms are well suited for a real-time smart e-Health home or a future robotic nurse. Finally, this paper is the first step for designing the ADMS for a smart e-Health home as the platform that can be extended for different healthcare policies such as screening process or diagnosing the disease class.

ACKNOWLEDGMENT

Also, the authors would like to thank Dr. Narjes Naghshvarianjahromi. She helps us to understand the medical doctors' decision-making approach better. Her recommendations were used in Section IV for the medical doctors' approach and the part related to the smoker in Section VI.C.1.

REFERENCES

[1] J. M. Fuster, *Cortex and Mind: Unifying Cognition*. Oxford, U.K.: Univ. Oxford, 2003.

[2] S. Feng, P. Setoodeh, and S. Haykin, "Smart home: Cognitive interactive people-centric Internet of Things," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 34–39, Feb. 2017.

[3] S. Haykin, *Cognitive Dynamic Systems: Perception-action Cycle, Radar and Radio*. Cambridge, U.K.: Univ. Cambridge, 2012.

[4] M. Naghshvarianjahromi, S. Kumar, and M. J. Deen, "Brain inspired dynamic system for the quality of service control over the longhaul non-linear fiber-optic link," *Sensors*, vol. 19, no. 9, pp. 2175–2195, 2019.

[5] M. Naghshvarianjahromi, S. Kumar, and M. J. Deen, "Brain inspired dynamic system for the quality of service control over the long-haul non-linear fiber-optic link," in *Proc. 16th Can. Workshop Inf. Theory (CWIT)*. Hamilton, ON, Canada: McMaster Univ., Jun. 2019, pp. 1–4.

[6] M. Naghshvarianjahromi, S. Kumar, and M. J. Deen, "Smart long-haul fiber optic communication systems using brain like intelligence," in *Proc. 16th Can. Workshop Inf. Theory (CWIT)*. Hamilton, ON, Canada: McMaster Univ., Jun. 2019, pp. 1–4.

[7] V. Gupta, S. Srinivasan, and S. S. Kudli, "Prediction and classification of cardiac arrhythmia," Stanford Univ., Stanford, CA, USA, Tech. Rep. css229, 2014.

[8] A. Hilmy, I. Syarif, and T. Badriyah, "Deep learning algorithm for arrhythmia detection," in *Proc. Int. Electron. Symp. Knowl. Creation Intell. Comput. (IES-KCIC)*, Sep. 2017, pp. 26–32.

[9] M. Karandikar and G. Guidi, "Classification of arrhythmia using ECG data," in *Proc. CS-Fall*, 2014, pp. 1–5.

[10] J. Henry, Y. Pylypchuk, T. Searcy, and V. Patel, "Adoption of electronic health record systems among us non-federal acute care hospitals: 2008-2015," *ONC Data Brief*, vol. 35, pp. 1–9, May 2016.

[11] M. Ghassemi, T. Naumann, F. Doshi-Velez, N. Brimmer, R. Joshi, A. Rumshisky, and P. Szolovits, "Unfolding physiological state: Mortality modelling in intensive care units," in *Proc. KDD*, 2014, pp. 75–84.

[12] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, R. Kim, R. Raman, P. C. Nelson, J. L. Mega, and D. R. Webster, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *J. Amer. Med. Assoc.*, vol. 316, no. 22, pp. 2402–2410, 2016.

[13] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, p. 115, 2017.

[14] Y.-A. Chung and W.-H. Weng, "Learning deep representations of medical images using siamese CNNs with application to content-based image retrieval," in *Proc. Mach. Learn. Health (ML4H) Workshop NIPS*, 2017, pp. 1–8.

[15] K. Nagpal, D. Foote, Y. Liu, P. H. C. Chen, E. Wulczyn, F. Tan, N. Olson, J. L. Smith, A. Mohtashamian, J. H. Wren, G. S. Corrado, R. MacDonald, L. H. Peng, M. B. Amin, A. J. Evans, A. R. Sangoi, C. H. Mermel, J. D. Hipp, and M. C. Stumpe, "Development and validation of a deep learning algorithm for improving gleason scoring of prostate cancer," *npj Digit. Med.*, vol. 2, no. 1, pp. 1–48, 2019.

[16] A. Raghu, M. Komorowski, L. A. Celi, P. Szolovits, and M. Ghassemi, "Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach," in *Proc. Mach. Learn. Healthcare (MLHC)*, 2017, pp. 1–17.

[17] W.-H. Weng, M. Gao, Z. He, S. Yan, and P. Szolovits, "Representation and reinforcement learning for personalized glycemic control in septic patients," in *Proc. Mach. Learn. Health (ML4H) Workshop NIPS*, 2017, pp. 1–5.

[18] M. Komorowski, L. A. Celi, O. Badawi, A. C. Gordon, and A. A. Faisal, "The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care," *Nature Med.*, vol. 24, no. 11, pp. 1716–1720, 2018.

[19] E. P. Lehman, R. G. Krishnan, X. Zhao, R. G. Mark, and L.-W. H. Lehman, "Representation learning approaches to detect false Arrhythmia alarms from ECG dynamics," in *Proc. Mach. Learn. Healthcare (MLHC)*, 2018, pp. 1–15.

[20] T.-M. H. Hsu, W.-H. Weng, W. Boag, M. McDermott, and P. Szolovits, "Unsupervised multimodal representation learning across medical images and reports," in *Proc. Mach. Learn. Health (ML4H) Workshop NeurIPS*, 2018, pp. 1–15.

[21] G. Liu, T.-M. H. Hsu, M. McDermott, W. Boag, W.-H. Weng, P. Szolovits, and M. Ghassemi, "Clinically accurate chest X-ray report generation," in *Proc. Mach. Learn. Healthcare (MLHC)*, 2019, pp. 1–20.

[22] S. Gehrmann, F. Dernoncourt, Y. Li, E. T. Carlson, J. T. Wu, J. Welt, J. Foote, Jr., E. T. Moseley, D. W. Grant, P. D. Tyler, and L. A. Celi, "Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives," *PLoS ONE*, vol. 13, no. 2, pp. 1–18, 2018.

[23] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[24] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, and S. Dieleman, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[25] E. Stav, S. Walderhaug, M. Mikalsen, S. Hanke, and I. Benc, "Development and evaluation of SOA-based AAL services in real-life environments: A case study and lessons learned," *Int. J. Med. Inform.*, vol. 82, pp. 269–293, Nov. 2013.

[26] S. Majumder, T. Mondal, and M. J. Deen, "Wearable sensors for remote health monitoring," *Sensors*, vol. 17, no. 1, p. 130, Jan. 2017.

[27] M. J. Deen, "Information and communications technologies for elderly ubiquitous healthcare in a smart home," *Pers. Ubiquitous Comput.*, vol. 19, nos. 3–4, pp. 573–599, 2015.

[28] H. Wang, N. Agoulmine, M. J. Deen, and J. Zhao, "A utility maximization approach for information-communication tradeoff in wireless body area networks," *Pers. Ubiquitous Comput.*, vol. 18, no. 8, pp. 1963–1976, 2014.

[29] H. Wang, H.-S. Choi, N. Agoulmine, M. J. Deen, and J. W.-K. Hong, "Information-based energy efficient sensor selection in wireless body area networks," in *Proc. IEEE Int. Conf. Commun.-Symp. Sel. Areas Commun. e-Health Track (ICC2011-SAC EH)*, Kyoto, Japan, Jun. 2011, pp. 1–6.

[30] S. Majumder, S. Majumder, E. Aghayi, M. Noferesti, H. Memarzadeh-Tehran, T. Mondal, Z. Pang, and M. J. Deen, "Smart homes for elderly healthcare-recent advances and research challenges," *Sensors*, vol. 17, no. 11, p. 2496, Oct. 2017.

[31] Autonomic Computing *et al.*, "An architectural blueprint for autonomic computing," 3rd ed., IBM Corp., Armonk, NY, USA, White Paper, Jun. 2005. [Online]. Available: https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf

[32] J. O. Kephart and D. M. Chess, "The vision of autonomic computing. Computer," *Comput.*, vol. 36, no. 1, pp. 41–50, Jan. 2003.

[33] N. F. Marko and R. J. Weil, "Non-Gaussian distributions affect identification of expression patterns, functional annotation, and prospective classification in human cancer genomes," *PLoS ONE*, vol. 7, no. 10, p. e46935, 2012. [Online]. Available: https://doi.org/10.1371/journal.pone.0046935

[34] R. Bono, M. J. Blanca, J. Arnau, and J. Gómez-Benito, "Non-normal distributions commonly used in health, education, and social sciences: A systematic review," *Frontiers Psychol.*, vol. 8, p. 1602, Sep. 2017.

[35] I. Perfilieva, J. Mockor, and V. Novák, *Mathematical Principles of Fuzzy Logic*. Dordrecht, The Netherlands: Kluwer Academic, 1999.

[36] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[37] R. E. Bellman and L. A. Zadeh, "Decision-making in a fuzzy environment," *Manage. Sci.*, vol. 17, no. 4, p. B-141, 1970.

[38] S. Voss, "Fuzzy logic in health care settings: Moral math for value-laden choices," *J. Humanistic Math.*, vol. 6, no. 2, pp. 161–178, 2016.

[39] J. H. T. Bates and M. P. Young, "Applying fuzzy logic to medical decision making in the intensive care unit," *Amer. J. Respiratory Crit. Care Med.*, vol. 167, no. 7, pp. 948–952, 2003.

[40] D. DeMazumder, D. E. Lake, A. Cheng, T. J. Moss, E. Guallar, R. G. Weiss, S. R. Jones, G. F. Tomaselli, and J. R. Moorman, "Dynamic analysis of cardiac rhythms for discriminating atrial fibrillation from lethal ventricular arrhythmias," *Circulat., Arrhythmia Electrophysiol.*, vol. 6, no. 3, pp. 555–561, 2013.

[41] A. Petrénas, L. Sörnmo, A. Lukoševičius, and V. Marozas, "Detection of occult paroxysmal atrial fibrillation," *Med. Biol. Eng. Comput.*, vol. 53, no. 4, pp. 287–297, 2015.

[42] M. Macgill. *What Should My Heart Rate be?* Accessed: Dec. 12, 2019. [Online]. Available: https://www.medicalnewstoday.com/articles/235710.php

[43] *Pediatric Tachycardia Algorithm (With a Pulse and Poor Perfusion)*. Accessed: Dec. 12, 2019. [Online]. Available: https://www.uptodate.com/contents/image?imageKey=EM%2F67438&topicKey=EM%2F6392&source=see_link

[44] T. Martin. *The Effects of Carbon Monoxide From Cigarette Smoking*. Accessed: Aug. 31, 2018. [Online]. Available: https://www.verywellmind.com/carbon-monoxide-in-cigarette-smoke-2824730

[45] P. Harris and D. Lysitsas, "Ventricular arrhythmias and sudden cardiac death," *BJA Educ.*, vol. 16, no. 7, pp. 221–229, 2016.

[46] R. Mehra, "Global public health problem of sudden cardiac death," *J. Electrocardiol.*, vol. 40, no. 6, pp. S118–S122, 2007.

[47] *Who is at Risk for an Arrhythmia?* Nat. Heart, Lung, Blood Inst., Bethesda, MD, USA, Jul. 2011.

[48] Brigham Women's Hospital. *Arrhythmia Treatment Services*. Accessed: Dec. 12, 2019. [Online]. Available: https://www.brighamandwomens.org/heart-and-vascular-center/resources/Arrhythmia-treatment

[49] S. S. Padhi, N. Patel, D. Driscoll, and B. Budgell, "Prevalence of cardiac arrhythmias in a community based chiropractic practice," *J. Can. Chiropractic Assoc.*, vol. 58, no. 3, pp. 238–245, 2014.

[50] S. M. Pehrson, C. Blomströ-Lundqvist, E. Ljungströ, and P. Blomströ, "Clinical value of transesophageal atrial stimulation and recording in patients with arrhythmia-related symptoms or documented supraventricular tachycardia—Correlation to clinical history and invasive studies," *Clin. Cardiol.*, vol. 17, no. 10, pp. 528–534, 1994.

[51] T. T. Hendrikx, M. Rosenqvist, P. Wester, H. Sandström, and R. Hörnsten, "Intermittent short ECG recording is more effective than 24-hour Holter ECG in detection of arrhythmias," *BMC Cardiovascular Disorders*, vol. 14, no. 1, pp. 14–41, 2014.

[52] Q. Zhang, X. Chen, Z. Fang, and S. Xia, "False arrhythmia alarm reduction in the intensive care unit using data fusion and machine learning," in *Proc. IEEE-EMBS Int. Conf. Biomed. Health Inform. (BHI)*, Las Vegas, NV, USA, Feb. 2016, pp. 232–235.

[53] *Matlab Sample Datasets*. Accessed: Dec. 12, 2019. [Online]. Available: https://www.mathworks.com/help/stats/sample-data-sets.html

[54] *UCI Machine Learning Repository*. Accessed: Dec. 12, 2019. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Arrhythmia

[55] *ECG Interpretation Part 1: Definitions Criteria and Characteristics of the Normal ECG (EKG) Waves, Intervals, Durations & Rhythm*. Accessed: Dec. 12, 2019. [Online]. Available: https://ecgwaves.com/ecg-normal-p-wave-qrs-complex-st-segment-t-wave-j-point/

[56] C. Luan and G. Dong, "Experimental identification of hard data sets for classification and feature selection methods with insights on method selection," *Data Knowl. Eng.*, vol. 118, pp. 41–51, Nov. 2018. [Online]. Available: http://www.sca-aware.org/sca-news/aha-releases-2015-heart-and-stroke-statistics

[57] A. Kumar. *Machine Learning: Validation Techniques*. Accessed: Dec. 12, 2019. [Online]. Available: https://dzone.com/articles/machine-learning-validation-techniques

[58] *The PhysioNet Computing in Cardiology Challenge 2017*. Accessed: Dec. 12, 2019. [Online]. Available: https://physionet.org/content/challenge-2017/1.0.0/

[59] S. Datta. *MATLAB Codes for Feature Extraction*. Accessed: Dec. 12, 2019. [Online]. Available: https://physionet.org/content/challenge-2017/1.0.0/sources/shreyasi-datta-209.zip

[60] M. Zabihi, A. B. Rad, A. K. Katsaggelos, S. Kiranyaz, S. Narkilahti, and M. Gabbouj, "Detection of atrial fibrillation in ECG hand-held devices using a random forest classifier," in *Proc. Comput. Cardiol. (CinC)*, Rennes, France, Sep. 2017, pp. 1–4.

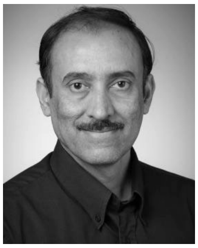
[61] S. Hong, M. Wu, Y. Zhou, Q. Wang, J. Shang, H. Li, and J. Xie, "ENCASE: An ENsemble ClASsifiEr for ECG classification using expert features and deep neural networks," in *Proc. Comput. Cardiol. (CinC)*, Rennes, France, Sep. 2017, pp. 1–4.

[62] S. Datta, C. Puri, A. Mukherjee, R. Banerjee, A. D. Choudhury, R. Singh, A. Ukil, S. Bandyopadhyay, A. Pal, and S. Khandelwal, "Identifying normal, AF and other abnormal ECG rhythms using a cascaded binary classifier," in *Proc. Comput. Cardiol. (CinC)*, Rennes, France, Sep. 2017, pp. 1–4.

[63] M. Zihlmann, D. Perekrestenko, and M. Tschannen, "Convolutional recurrent neural networks for electrocardiogram classification," in *Proc. Comput. Cardiol. (CinC)*, Rennes, France, Sep. 2017, pp. 1–4.

[64] Z. Xiong, M. K. Stiles, and J. Zhao, "Robust ECG signal classification for detection of atrial fibrillation using a novel neural network," in *Proc. Comput. Cardiol. (CinC)*, Rennes, France, Sep. 2017, pp. 1–4.

MAHDI NAGHSHVARIANJAHROMI received the M.S. degree in communications engineering from the Amirkabir University of Technology (AUT), in 2014. He is currently pursuing the Ph.D. degree in electrical and computer engineering with McMaster University. He published more than 31 articles. His current research interests include fiber-optic communications systems and nonlinear impairment mitigation, smart home and automatic healthcare systems, computer sciences, statistical processing and brain-like intelligence, microwave imaging, cognitive dynamic system antennas, arrays, and RF devices.



SHIVA KUMAR received the M.S. and Ph.D. degrees in electrical communication engineering from the Indian Institute of Science, Bengaluru, India, in 1990 and 1994, respectively, and the Ph.D. degree in communications engineering from Osaka University, Osaka, Japan, in 1997. He worked at Corning, Inc., New York, as a Senior Research Scientist from 1998 to 2001, as a Supervisor from 2001 to 2002 and then as a Research Associate from 2002 to 2003. In 2003, he joined McMaster University, ON, Canada, where he is currently a Professor. He has authored a textbook on fiber optic communications (2014), edited a book on nonlinear fiber optics (2011), and has published about 80 articles in refereed journals and seven book chapters.

M. JAMAL DEEN is currently a Distinguished University Professor with McMaster University, a Senior Canada Research Chair in information technology, and the Past President of the Academy of Science, Royal Society of Canada (RSC). His research records include 560 peer-reviewed articles and two textbooks *Silicon Photonics—Fundamentals and Devices* and *Fiber Optic Communications-Fundamentals and Applications*. His current research interests include nano-/opto-electronics, nanotechnology, and their emerging applications in health and environment. He was elected to fellow status in ten national academies and professional societies, including RSC, APS, ECS, and AAAS.