UNIVERSIDADE ESTADUAL DO CEARÁ
CENTRO DE CIÊNCIAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

AFONSO BEZERRA LIMA NETO

A MULTI-AGENT SYSTEM USING FUZZY LOGIC APPLIED TO E-HEALTH IN ORDER TO MONITOR HYPERTENSION

FORTALEZA – CEARÁ
2018

AFONSO BEZERRA LIMA NETO

A MULTI-AGENT SYSTEM USING FUZZY LOGIC APPLIED TO E-HEALTH IN ORDER TO MONITOR HYPERTENSION

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Marcial Porto Fernandez

Co-Orientador: Gustavo Augusto Lima de Campos

FORTALEZA – CEARÁ

2018

Dados Internacionais de Catalogação na Publicação

Universidade Estadual do Ceará

Sistema de Bibliotecas

Lima Neto, Afonso Bezerra.
A multi-agent system using fuzzy logic applied to e-health in order to monitor hypertension [recurso eletrônico] / Afonso Bezerra Lima Neto. - 2018.
1 CD-ROM: il.; 4 ¾ pol.

CD-ROM contendo o arquivo no formato PDF do trabalho acadêmico com 97 folhas, acondicionado em caixa de DVD Slim (19 x 14 cm x 7 mm).

Dissertação (mestrado acadêmico) - Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Mestrado Acadêmico em Ciência da Computação, Fortaleza, 2018.
Área de concentração: Ciência da Computação.
Orientação: Prof. Dr. Marcial Porto Fernandez.
Coorientação: Prof. Dr. Gustavo Augusto Lima de Campos.

1. Lógica Fuzzy. 2. Sistema Multi-agente. 3. Saúde. 4. IoT. 5. Multi-agent system. I. Título.

AFONSO BEZERRA LIMA NETO

A MULTI-AGENT SYSTEM USING FUZZY LOGIC APPLIED TO E-HEALTH IN ORDER TO MONITOR HYPERTENSION

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em: 21 de Setembro de 2018

BANCA EXAMINADORA

Marcial Porto Fernandez (Orientador)
Universidade Estadual do Ceará – UECE

Gustavo Augusto Lima de Campos (Co-Orientador)
Universidade Estadual do Ceará – UECE

Rafael Lopes Gomes
Universidade Estadual do Ceará – UECE

Francisco Nauber Bernardo Gois
Universidade Federal do Ceará - UFC

To my family, for their ability to believe in me and invest in me. My wife Claudia, my children João Victor and Ana Beatriz, their understanding and patience were that they gave, in some moments, the hope to follow. My sister Júlia also gave me support to start and continue on this journey.

ACKNOWLEDGEMENTS

First to God by the force and protection given throughout my life, principally in the moments of difficulties and falls, he held me with his hand and raised me to go on.

To my beloved wife Claudia who has always been by my side at all times of this journey, supporting and encouraging me to move on. To my beloved children, João Víctor and Ana Beatriz, for their patience on moments of absence. To our beloved Lilo for the joy, he brought to our family in such a suffering moment that we were passing.

To my beloved father Antonio Augusto (in memorian) and my beloved mother Maria Júlia, for love, encouragement and unconditional support throughout my life. The example of dedicated university professors and professionals loved by their UECE and UFC students and colleagues, as well as an example of a person with unquestionable conduct and integrity, have always shown me that study, serious work, love and dedication to the family can walk together and generate great fruits.

Thanks to my sisters Julia and Regina for the support given and words of encouragement. With the initial support of Julia, this project became a reality.

I would like to thank nurses Francisco Alexsandro Sousa Rocha and Márcia Cymara Marinho Jardins for the support in the supply of the input parameters according to the current valid rules, as well as in the validation of the presented inferences.

To this university, its faculty, direction, and administration, which have made it possible for the window, which today I see a higher horizon, amidst the fervent confidence in the merit and ethics present here.

I thank all the teachers for providing me with not only rational knowledge but the manifestation of the character and affectivity of education in the process of a professional formation so that they dedicated themselves to me, not only for having taught me but for having made me learn.

Special thanks to Professor Jorge Luiz for listening to me at the beginning of everything and showing me the first course to follow. To Professor Paulo Henrique also for showing me that the search for new knowledge can be something pleasurable. To Professor Gustavo Campos for making artificial intelligence an exciting subject to study and learn and to Professor Marcial Fernandez for the patience and guidance on this work.

"Cast all your anxiety on the Lord, because He cares for you."

(1 Peter, 5-7)

RESUMO

Com o passar do tempo, a qualidade de vida está se tornando uma das maiores preocupações das pessoas que estão envelhecendo. De acordo com estudos envolvendo países da Europa, pessoas mais velhas tendem a viver sozinhas ou, no máximo, com uma outra pessoa na mesma casa. A tecnologia atualmente disponível, considerando produtos de saúde, ajuda essas pessoas a alcançarem seus objetivos. Contudo, as soluções que contém essa tecnologia ainda tem um caráter muito genérico e proprietário de cada fabricante, impedindo a integração com outras soluções com arquitetura aberta, de forma a gerar uma aproveitamento maior dos dados capturados e identificar problemas a partir de outras perspectivas. Levando isso em conta, este trabalho propõe uma arquitetura de sistemas multiagente que utiliza dispositivos de IoT para captar sinais cardíacos de pacientes e, utilizando a inteligência artificial através da lógica fuzzy, estimar o nível de hipertensão considerando pressão sistólica, pressão diastólica, idade e índice de massa corporal. Utilizamos informações de 768 pacientes obtidos de um banco de dados público e avaliamos o desempenho do modelo de lógica fuzzy apresentado. A solução proposta comparou os resultados dessa lógica fuzzy com uma avaliação feita por enfermeiros, atingindo uma precisão de 94,40% no diagnóstico.

Palavras-chave: Lógica Fuzzy. Sistema Multi-agente. Saúde. IoT.

ABSTRACT

As time is passing by, life quality is becoming one of the most concerns for people who are getting old. According to studies involving countries in Europe, older people tend to live alone or, at most, with another person. The technology, currently available considering health products, helps those people to achieve their goals. However, the solutions that contain this technology still have a generic and proprietary character of each manufacturer, not allowing the integration with other solutions with open architecture, in order to generate a better use of the captured data, and to identify problems from other perspectives. Taking that into account, this work proposes a multi-agent system architecture that uses IoT devices to catch patients' heart signals and, using artificial intelligence through fuzzy logic process to estimate the level of hypertension, considering systolic pressure, diastolic pressure, age, and body mass index. We used information about 768 patients obtained from a public database and evaluated the performance of the presented fuzzy logic model. The proposed solution compared the results of such fuzzy logic with an evaluation made by nurses, reaching a 94.40% of accuracy in the diagnosis.

Keywords: Fuzzy logic. Multi-agent system. Health. IoT.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF SOURCE CODE

LIST OF ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface
BMI	Body Mass Index
BP	Blood Pressure
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
DBP	Diastolic Blood Pressure
ECG	Electrocardiogram
IoT	Internet of Things
LCD	Liquid Crystal Display
MAS	Multi-Agent System
MIoT	Medical Internet of Things
MQTT	Message Queuing Telemetry Transport
NoSQL	not only SQL
PPG	Photoplethysmography
RDBMS	Relational Database Management System
SBP	Systolic Blood Pressure
SQL	Structured Query Language

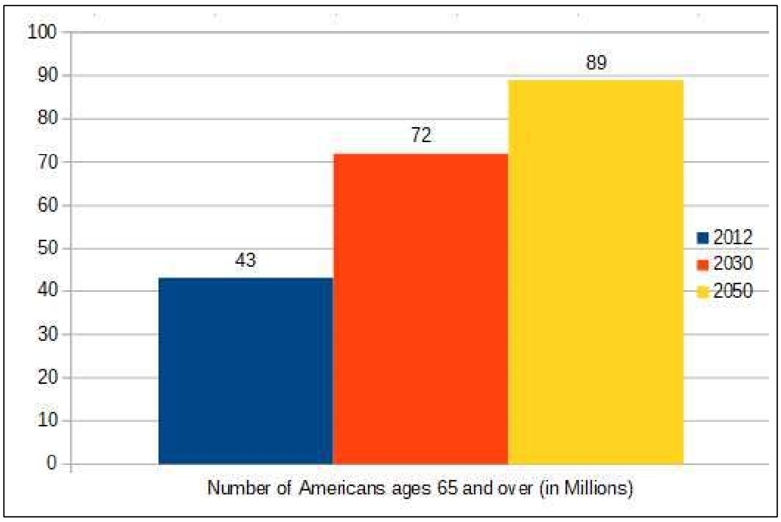
CONTENTS

1 INTRODUCTION

In 2050, world population may rise nearly 20% when compared with 2017, as published by the European Commission Research and Innovation report (COMMISSION, 2014), having the majority living in small households - alone or with just one person. Everyone desires to have an optimal quality of life, and according to their financial condition, invest money in products and services related to health.

In developed countries, such as the USA, there are benefits from sweeping advances in nutrition, sanitation, and medicine that transformed public health practice and increased average lifespan during the first half of the twentieth century (AGING, 2016). Figure 1 depicts the evolution of American's expectancy of life, where in 2012 there were 43 million Americans with 65 years or more, and in 2050 the expectancy is an increase of approximately 100%.

Figure 1 – Evolution of Americans' expectance of life



Source – Elaborated by the author

In the last decade, wearable devices attracted much attention from the academic community and industry, becoming recently very popular. The most relevant definition of wearable electronics is the following: "devices that can be worn or mated with human skin to continuously and closely monitor an individual's activities, without interrupting or limiting the user's motions" (HAGHI; THUROW; STOLL, 2017). Currently, mobile apps and wearable devices have been integrated with medicine purposes, structuring the medical internet of things. Application areas for the IoT reside on medical and healthcare (PANG, 2013). The new concept of Medical Internet of Things (MIoT) involves wearable devices that monitor vital signals in real time. Interconnection of sensors and medical devices has been the goal of many industry players, considering that this is part of the core of Internet of Things (IoT). This research has been conducted to provide an environment in which technology can improve people's health and welfare.

1.1 MOTIVATION

Blood Pressure (BP) measurements are usually used to diagnose Hypertension. Casual BP measurements are of limited effectiveness because they do not reflect the circadian variation in BP, the "white-coat effect," regression to the mean, observer bias, and other factors (IMAI *et al.*, 1997). Other factors that contribute to the variability of blood pressure, such as sex, age, and Body Mass Index (BMI).

The most of devices that catch vital signals act in an isolated way and the gathered data reach a limited level of eficacy to identify some health problem of the user. The processing of those data to generate some useful information, in the most of cases, uses technologies that don't consider all possible scenarios, due to some conceptual limitations and implementation.

Considering the lack of knowledge of most people in perceiving any imminent cardiac problems, and using technology in the sake of improve the way the separated parts of the system can interact and help each other to achieve a global goal, in this work is proposed a solution with Multi-Agent System (MAS), involving pattern of information exchange and artificial intelligence techniques, that sends information of heart signals using IoT, processes the data remotely, and infers diagnosis of high blood pressure, showing the result on a remote device. The solution proposed uses fuzzy logic, embedded into that multi-agent system. Fuzzy logic has a better performance when compared with other artificial intelligence approaches. To evaluate the performance of the presented fuzzy logic model was used information from 768 patients obtained from a public database. The results of such fuzzy logic system were compared with an evaluation made by experts.

1.2 CONTRIBUTION

The contribution of the present work generated researches in fields related to multi-agent systems, artificial intelligence, and internet of things. Thus, it was proposed a Multi-Agent architecture to infer diagnoses in hypertensive patients, which was presented in (NETO *et al.*, 2018b) and (NETO *et al.*, 2018a).

1.3 OBJECTIVES

1.3.1 GENERAL OBJECTIVE

This work proposes a MAS applied for e-Health, using IoT devices to monitor individuals heart signals, applying fuzzy logic to infer the level of their blood pressure, and alerts someone to help or guide them in first aids.

1.3.2 Specific Objectives

For the main objective to be achieved, the following specific objectives were established:

a) Define agents roles - mobile, monitoring and processing - with specific responsibilities.
b) Present an architecture to integrate all agents.
c) Process the information, applying fuzzy logic and infer diagnosis of the level of blood pressure.
d) Monitor the levels of blood pressure and alert when they reach a severe degree.

1.4 OUTLINE

The remaining text of this dissertation is organized as follow: Chapter 2 presents the background. Chapter 3 shows related works that supported this work and the state of the art considering e-Health. Chapter 4 depicts the MAS solution proposed in this work. Chapter 5 details the evaluation and results obtained with the present MAS solution, and Chapter 6 describes conclusion and future works.

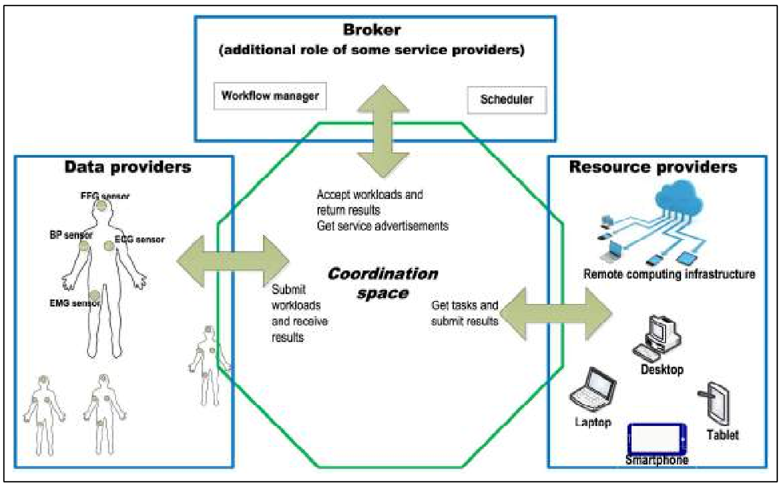
2 BACKGROUND

This chapter presents the basic concepts for understanding the proposal considering multi-agent systems applied to e-Health.

2.1 IOT HEALTHCARES NETWORK

One of the essential elements of the IoT in healthcare is the IoT Healthcare Network (IoThNet) (ISLAM *et al.*, 2015). It provides access to the IoT backbone, facilitates medical data transmission and reception, and enables the use of healthcare-tailored communications. The way in which sign data are gathered using sensors, such as Electrocardiogram (ECG), Blood Pressure, etc., and how IoThNet topology distributes, processes and visualizes those data are depicted in Figure 2.

Figure 2 – A conceptual diagram of IoT-based ubiquitous healthcare solutions



Source – (ISLAM *et al.*, 2015)

The Broker controls when (scheduler) and how (workflow manager) the data is transmitted through the environment. The Resource Providers are responsible for giving all support to keep the system working fine and stable, as well as allow that information to be accessed by several devices, such as tablet or smartphone. The Coordination Space manages several requests and guides them on the right flow. Finally, several kinds of sensors (ECG/ PPG) connected to the body represents the Data Providers to get vital signals and send them to processing.

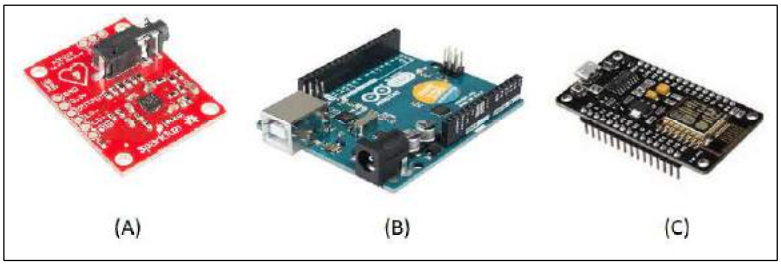
2.2 SENSORS, CONTROLLERS AND MOBILE APP

To get the electrical activity from the heart over a period is necessary using electrodes placed on the skin, and a shield connected to those electrodes over wires. There are several types of shields available on the market, such as the AD8232 Heart Rate Monitor Hookup (SPARKFUN, 2015), to provide ECG. That hardware works with an Arduino or NodeMCU board.

Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) which runs on a computer and is used to write and upload computer code to the physical board. NodeMCU is an open source IoT platform that includes firmware which runs on a Wi-Fi environment.

There are mobile apps that get information as well as and other devices that send that information to another place. Figure 3 shows these devices.

Figure 3 – Physical devices for IoT.



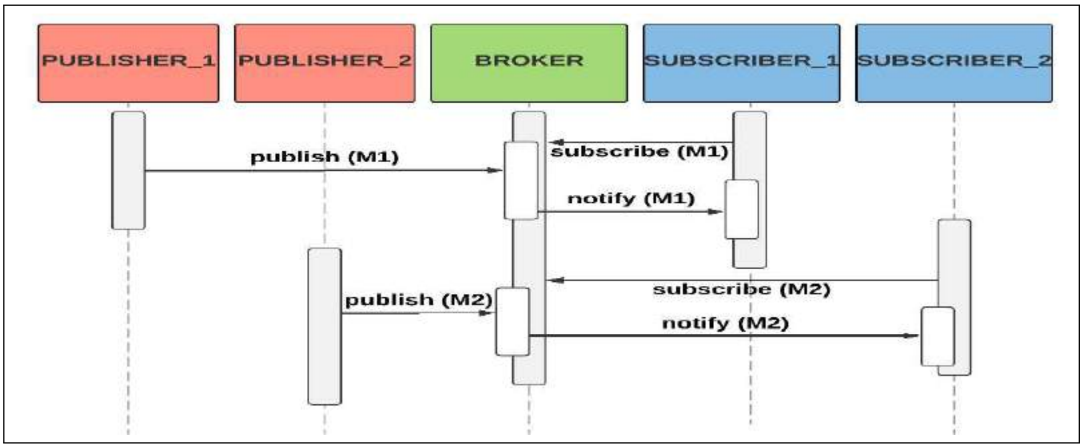
Source – Elaborated by the author

The device (A) is a AD8232 board specifically developed to connect ECG sensors, capture the signals and send through an Arduino board, for example. The device (B) is an Arduino board to get data from others boards, such as device (A). The device (C) is a NodeMCU, which in the same way as the arduino, can have other devices connected. The difference is that the NodeMCU already comes with built-in Wi-Fi.

2.3 PUBLISH/SUBSCRIBE PATTERN

The publish/subscribe pattern is part of a message pattern, which describes how two different parts of a message passing system connect and communicate with each other. As shown in Figure 4, the publishers do not program the messages to be sent directly to specific subscribers, but instead categorize published messages into classes on a broker, without knowledge of which subscribers. Any message published to a topic on the broker is immediately received by all the subscribers to the topic.

Figure 4 – Publish/Subscribe Pattern



Source – Elaborated by the author

In modern cloud architecture, applications are decoupled into smaller, independent building blocks that are easier to develop, deploy and maintain. Publish/Subscribe messaging provides instant event notifications for these distributed applications. Considering that e-Health applications involves several devices (hardware or software) that should communicate to each other, is possible do see the adherence between e-Health and this pattern.

The Publish/Subscribe model allows messages to be broadcast to different parts of a system asynchronously. A sibling to a message queue, a message topic provides a lightweight mechanism to broadcast asynchronous event notifications and endpoints that allow software components to connect to the topic to send and receive those messages. To broadcast a message, a component called a publisher pushes a message to the topic. Unlike message queues, which batch messages until they are retrieved, message topics transfer messages with no or very little queuing and push them out immediately to all subscribers. All components that subscribe to the topic will receive every message that is broadcast unless the subscriber sets a message filtering policy.

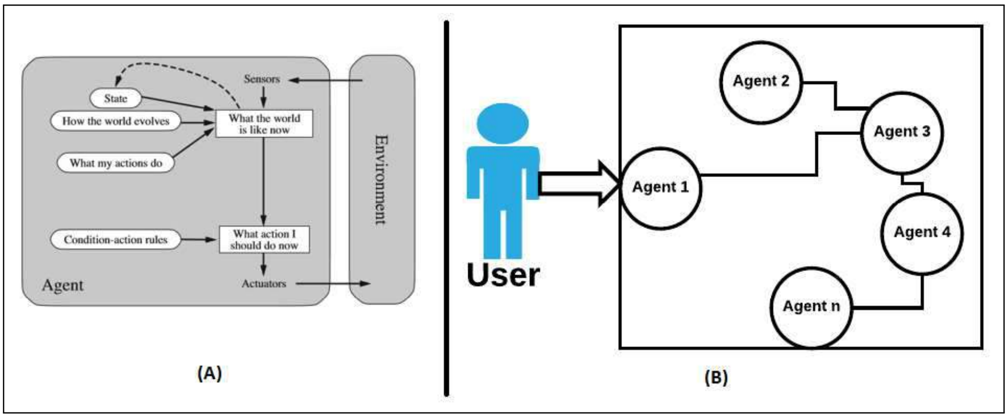
The subscribers to the message topic often perform different functions, and can each do something different with the message in parallel. The publisher doesn't need to know who is using the information that it is broadcasting, and the subscribers don't need to know from who the message comes. This style of messaging is a bit different than message queues, where the component that sends the message often knows the destination it is sending to.

The broker typically performs a store and forward function to route messages from publishers to subscribers. Besides, the broker may prioritize messages in a queue before routing.

2.4 MULTI-AGENT SYSTEMS (MAS)

In artificial intelligence research, agent-based systems technology has been hailed as a new paradigm for conceptualizing, designing, and implementing software systems. According to Russel and Norving (RUSSELL; NORVIG, 2010), an agent is “anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.”. Figure 5 shows the elements of an agent and a Multi-Agent System.

Figure 5 – (A) Elements of an agent and (B) Multi-Agent System(MAS).



Source – Elaborated by the author

As shown in Figure 5.A, an agent can handle partially observable environments. Its current state is stored inside the agent maintaining some structure which describes the part of the world which cannot be seen. The knowledge about "how the world works" is called a model of the world.

Depending on the agent, it should maintain some internal model that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state. Percept history and impact of an action on the environment can be determined by using an internal model. It then chooses an action according to the rules defined previously.

Increasingly, as shown in Figure 5.B, multiple agents that can work together are required by applications. A MAS is a network of software agents that are loosely coupled and interact to solve problems that are beyond the individual capacities or knowledge of each problem solver (SHOHAM; LEYTON-BROWN, 2010).

There are several types of agents. Each of them follows the respective agent program, such as reflex agents, whom the function is shown on the Algorithm 1. It acts according to a rule whose condition matches the current state, as defined by the percept.

Algorithm 1: Simple Reflex Agent

Input: percept

Output: an action

persistent: rules, a set of condition–action rules

state ← INTERPRET-INPUT(percept)

rule ← RULE-MATCH(state, rules)

action ← rule.ACTION

return *action*

Simple reflex agents have the admirable property of being simple, but they turn out to be of limited intelligence. The agent in Algorithm 1 will work only if the correct decision can be made on the basis of only the current percept, that is, only if the environment is fully observable. Even a little bit of unobservability can cause serious trouble.

Considering an environment is described as the particular situation in which an agent is present. Agents do generally not have complete control over their environments but more likely to have partial control. This means that they can influence the environment. In turn, changes to the environment will directly affect the agents in the environment.

Environments are generally said to be non-deterministic. Actions performed in certain environments can fail. Also, an agent performing the same task in two similar environments may result in entirely different outcomes.

An agent in an environment will have a pre-programmed set of abilities that it can apply to different situations that it may come across in its environment. From the diagram on agents, is possible to see that the agent has a sensor that is affected by the environment. The agent can use information from this sensor along with information already known to decide which action to perform. Apparently, not all steps can be performed in every situation. For example, an agent may have the ability to ‘open door’ but can only do so if the door is unlocked. The fact the door must be unlocked before the action can be performed is called a precondition.

The biggest problem associated with agents in an environment is deciding which action to perform at a given instance to maximize its chance of completing its goal or at least working towards it. The complexity of the decision-making process is affected by the properties of an environment.

Russell and Norvig (RUSSELL; NORVIG, 2010) suggest the following environment properties:

- Accessible/inaccessible: This refers to the possibility for an agent to obtain complete and accurate information about the environment's state. Examples: (I) Inaccessible environment => physical world => information about any event on earth; (II) Accessible environment => empty room which state is defined by its temperature and agents can measure it.
- Determinism/non-determinism: In a deterministic environment any action has a single guaranteed effect, and no failure or uncertainty. On the contrary is a non-deterministic environment. In this environment, the same task performed twice may produce different results or may even fail. Examples: (I) Non-deterministic environment => physical world: Robot on Mars; (II) Deterministic environment => Tic Tac Toe game.
- Episodic/non-episodic: In an episodic environment, each agent's performance is the result of a series of independent tasks performed. There is no link between the agent's performance and other different scenarios. In other words, the agent decides which action is best to take. It will only consider the task at hand and doesn't have to consider the effect it may have on future tasks. Examples: (I) Episodic environment => mail sorting system; (II) Non-episodic environment => chess game
- Static/Dynamic: An environment is static if only the actions of an agent modify it. It is dynamic on the other hand if other processes are operating on it. Examples: (I) Dynamic environment => physical world; (II) Static environment => empty office with no moving objects.
- Discrete/Continuous: An environment is said to be discrete if there are a finite number of actions that can be performed within it. Examples: (I) Discrete environment => A game of chess or checkers where there are a set number of moves; (II) Continuous environment => Taxi driving. There could be a route from to anywhere to anywhere else.

The more complex an environment is, the harder it is to decide which action to perform. The most complex environment is one that is inaccessible, non-deterministic, non-episodic, dynamic and continuous. However, in this work due to the features involved match a simple reflex agent, was considered the most simple environment: accessible - there is complete information about the environment; deterministic - the rules used by the agent has specific results; episodic - each processing task is independent from another; static - only agents can modify the environment; and discrete - there is a finite number of rules to be used by the agent.

2.5 FUZZY LOGIC

The context of the theory of fuzzy sets, presented by Zadeh (ZADEH, 1965) was used to introduce Fuzzy logic. A fuzzy set is a class of objects with a continuum of grades of membership. Such a set is characterized by a membership (characteristic) function which assigns to each object a degree of membership ranging between zero and one. The notions of inclusion, union, intersection, complement, relation, convexity, etc., are extended to such sets, and various properties of these notions in the context of fuzzy sets are established.

While variables usually take numerical values in mathematics, in fuzzy logic applications, non-numeric values are often used to facilitate the expression of rules and facts. A linguistic variable such as height may accept values such as tall and its antonym short. Because natural languages do not always contain enough value terms to express a fuzzy value scale, it is common practice to modify linguistic values with adjectives or adverbs (TAMIR; RISHE; KANDEL, 2015).

Membership functions are used to quantify linguistic term and represent a fuzzy set graphically. A membership function for a fuzzy set A on the universe of discourse X is defined as

$$\mu A : X \rightarrow [0,1] \tag{2.1}$$

Membership functions map each element of X and give them a value between 0 and 1. It is called membership value or degree of membership and quantifies the degree of membership of the element in X to the fuzzy set A. They have the following features, as depicted in Figure 6:

- Core - For any fuzzy set A, the core of a membership function is that region of universe that is characterize by full membership in the set. Hence, core consists of all those elements y of the universe of information such that,

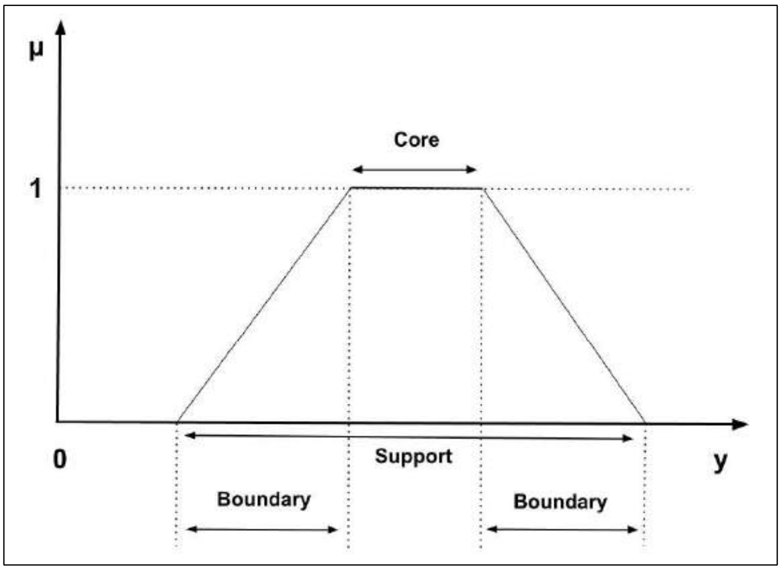
$$\mu A(y) = 1 \tag{2.2}$$

- Support - For any fuzzy set A, the support of a membership function is the region of universe that is characterize by a nonzero membership in the set. Hence core consists of all those elements y of the universe of information such that,

$$\mu A(y) > 0 \tag{2.3}$$

- Boundary - For any fuzzy set A, the boundary of a membership function is the region of universe that is characterized by a nonzero but incomplete membership in the set. Hence,

Figure 6 – Features of Membership Function.



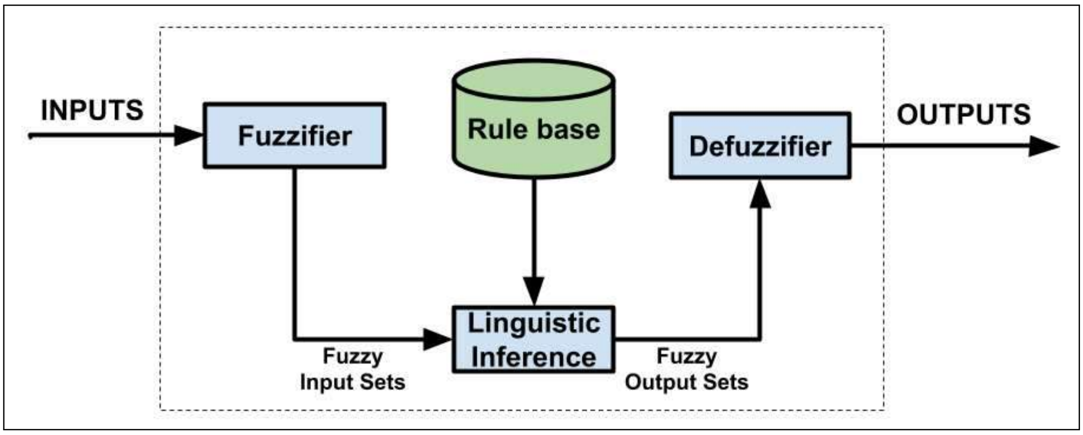
Source – tutorialspoint.com

core consists of all those elements y of the universe of information such that,

$$1 > \mu A(y) > 0 \tag{2.4}$$

As shown in Figure 7, the fuzzy logic process consists of four main steps:

Figure 7 – Fuzzy Logic Overview.



Source – Elaborated by the author

- Fuzzifier Module: It transforms the system inputs, which are crisp numbers, into fuzzy sets. Following are the two important methods of fuzzification:
  - Support Fuzzification(s-fuzzification) Method: In this method, the fuzzified set can be expressed with the help of the following relation

$$A = \mu_1 Q(x_1) + \mu_2 Q(x_2) + ... + \mu_n Q(x_n) \tag{2.5}$$

Here the fuzzy set Q(xi) is called as kernel of fuzzification. This method is implemented by keeping μi constant and xi being transformed to a fuzzy set Q(xi).

  - Grade Fuzzification (g-fuzzification) Method: It is quite similar to the above method but the main difference is that it kept xi constant and μi is expressed as a fuzzy set.

- Rule Base: It stores IF-THEN rules provided by experts.
- Linguistic Inference: It simulates the human reasoning process by making a fuzzy inference on the inputs and IF-THEN rules. There are two important methods of FIS, having different consequent of fuzzy rules Mamdani Fuzzy Inference System and Takagi-Sugeno Fuzzy Model (TS Method).
  - Mamdani Fuzzy Inference System - This system was proposed in 1975 by Ebhasim Mamdani. Basically, it was anticipated to control a steam engine and boiler combination by synthesizing a set of fuzzy rules obtained from people working on the system. Steps for computing the output:
    1. Set of fuzzy rules need to be determined in this step;
    2. In this step, by using input membership function, the input would be made fuzzy;
    3. Now establish the rule strength by combining the fuzzified inputs according to fuzzy rules;
    4. In this step, determine the consequent of rule by combining the rule strength and the output membership function;
    5. For getting output distribution combine all the consequents;
    6. Finally, a defuzzified output distribution is obtained.
  - Takagi-Sugeno Fuzzy Model (TS Method) - This model was proposed by Takagi, Sugeno and Kang in 1985. Format of this rule is given as

$$\text{IF x is A and y is B THEN Z = f(x,y)} \tag{2.6}$$

    Here, AB are fuzzy sets in antecedents and z = f(x,y) is a crisp function in the consequent. The fuzzy inference process under Takagi-Sugeno Fuzzy Model (TS Method) works following steps below:
    1. Fuzzifying the inputs: here, the inputs of the system are made fuzzy;
    2. Applying the fuzzy operator: in this step, the fuzzy operators must be applied to get the output.

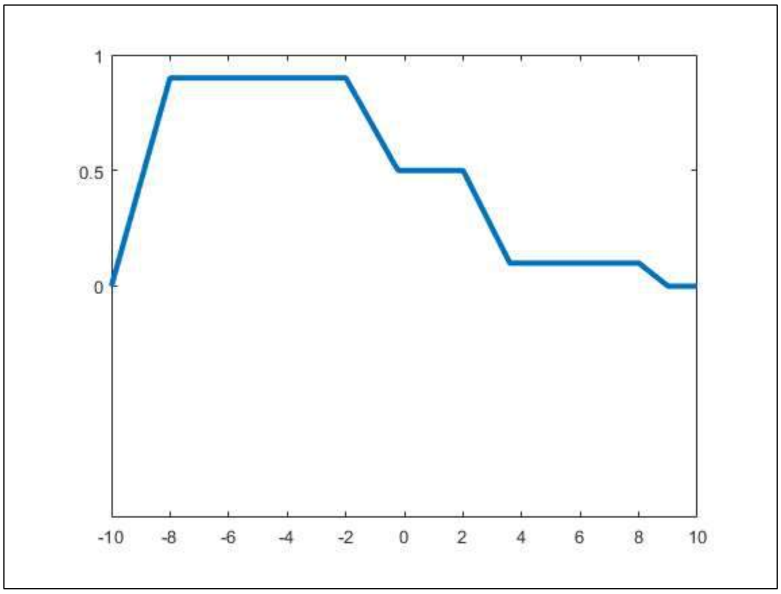
  Comparing these two methods, we have:
  - Output Membership Function: the main difference between them is on the basis of

output membership function. The Sugeno output membership functions are either linear or constant;

- Aggregation and Defuzzification Procedure: the difference between them also lies in the consequence of fuzzy rules and due to the same their aggregation and defuzzification procedure also differs;
- Mathematical Rules: more mathematical rules exist for the Sugeno rule than the Mamdani rule;
- Adjustable Parameters: the Sugeno controller has more adjustable parameters than the Mamdani controller.

• Defuzzifier Module: It transforms the fuzzy set obtained by the inference engine into a crisp value. There are many different methods of defuzzification available, such as BADD (Basic Defuzzification Distributions), FCD (Fuzzy Clustering Defuzzification), Centroid, Bisector, MOM (Middle of Maximum), SOM (Smallestof Maximum), and LOM (Largest of Maximum). To show how defuzzification works, suppose we have the following region to be defuzzified, as shown in Figure 8. Taking the methods Centroid, Bisector, LOM, SOM and MOM, which one would be the best?

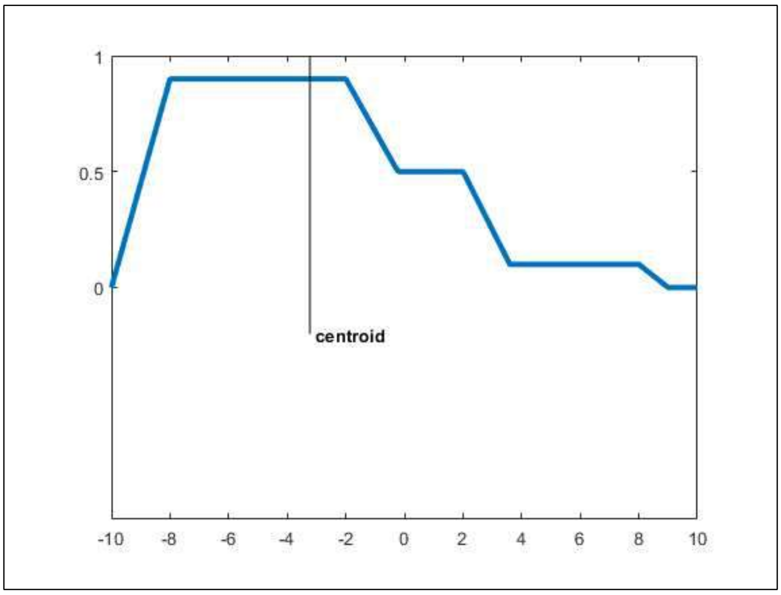
Figure 8 – Region to be defuzzified.



Source – mathworks.com

- Centroid: returns the center of area under the curve. If you think of the area as a plate of equal density, the centroid is the point along the x axis about which this shape would balance (see Figure 9).

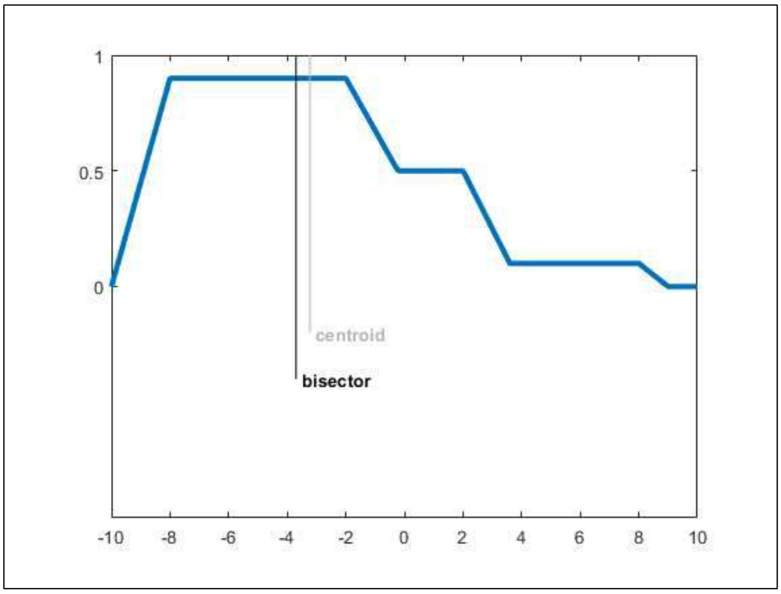
Figure 9 – Defuzzification using Centroid.



Source – mathworks.com

- Bisector: the bisector is the vertical line that will divide the region into two sub-regions of equal area. It is sometimes, but not always coincident with the centroid line (see Figure 10).

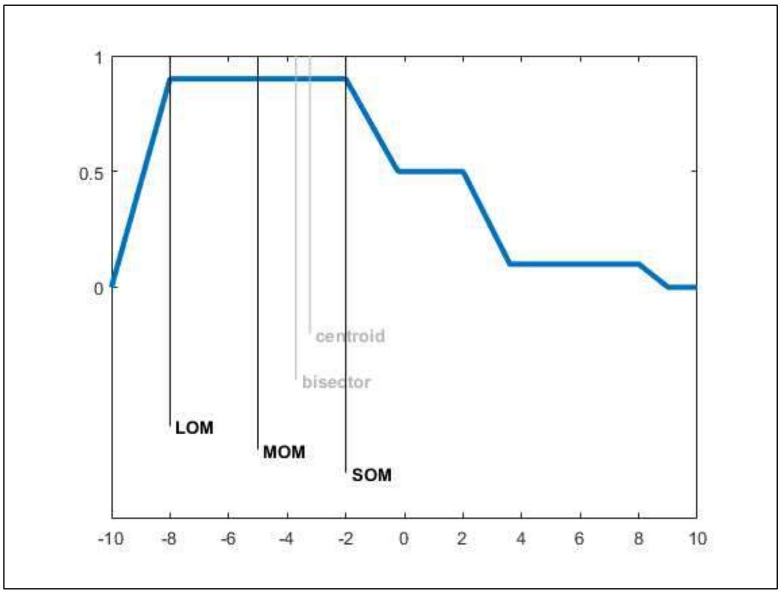
Figure 10 – Defuzzification using Bisector.



Source – mathworks.com

- LOM, SOM and MOM: These three methods key off the maximum value assumed by the aggregate membership function. In this example, because there is a plateau at the maximum value, they are distinct. If the aggregate membership function has a unique maximum, then MOM, SOM, and LOM all take on the same value (see Figure 11).

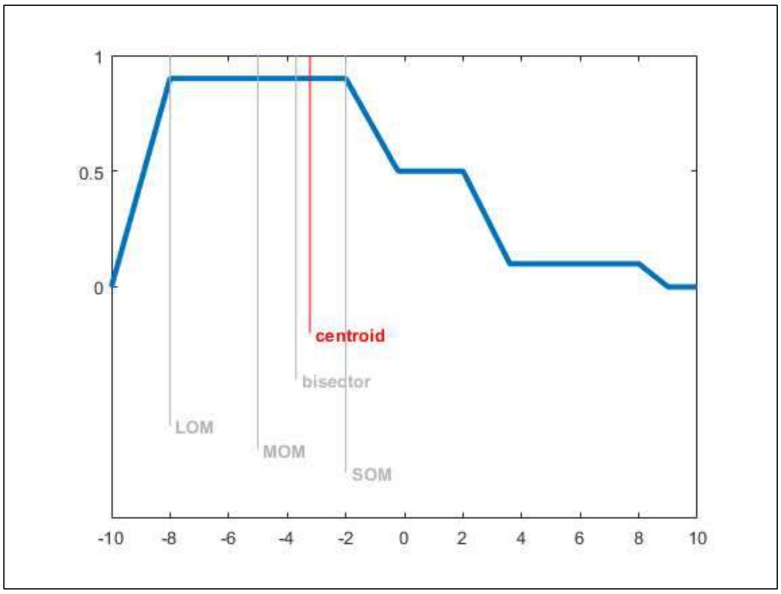
Figure 11 – Defuzzification using LOM, SOM and MOM.



Source – mathworks.com

Which of these methods is the right one? There's no simple answer. To get started quickly, generally the centroid method is good enough. Later is possible change the defuzzification method to see if another method works better (see Figure 12).

Figure 12 – Picking a Method.



Source – mathworks.com

2.6 HIGH BLOOD PRESSURE

High blood pressure (HBP or hypertension) is the force exerted by the blood against the walls of the blood vessels. The pressure depends on the work being done by the heart and the resistance of the blood vessels.

Medical guidelines define hypertension as a blood pressure higher than 130 over 80 millimeters of mercury (mmHg), according to guidelines issued by the American Heart Association (AL-KHATIB *et al.*, 2017).

Hypertension and heart disease are global health concerns. The World Health Organization (WHO) suggests that the growth of the processed food industry has impacted the amount of salt in diets worldwide and that this plays a role in hypertension.

2.6.1 Key Points

Key points about hypertension are the following:

- Normal blood pressure is 120 over 80 mm of mercury (mmHg), but hypertension is higher than 130 over 80 mmHg;
- Acute causes of high blood pressure include stress, but it can happen on its own, or it can result from an underlying condition, such as kidney disease;
- Unmanaged hypertension can lead to a heart attack, stroke, and other problems;
- Lifestyle factors are the best way to address high blood pressure.

2.6.2 Risk Factors

In most cases, it's not clear exactly what causes high blood pressure (hypertension). But several things that can increase the risk of a heart problem. Furthermore, factors that can raise the risk of developing high blood pressure include:

- Age – the risk of developing high blood pressure increases as you get older;
- A family history of high blood pressure;
- Ethnicity - Some ethnic groups are more prone to hypertension;
- A salt-rich diet associated with processed and fatty foods;
- Lack of exercise;
- Size and weight: Being overweight or obese is a key risk factor;
- Regularly drinking large amounts of alcohol;

- Smoking;
- Long-term sleep deprivation.

2.6.3 Symptoms

A person with hypertension may not notice any symptoms, and it is often called the "silent killer." While undetected, it can cause damage to the cardiovascular system and internal organs, such as the kidneys.

Regularly checking blood pressure is vital, as there will usually be no symptoms to make people aware of the condition. It is maintained that high blood pressure causes sweating, anxiety, sleeping problems, and blushing. However, in most cases, there will be no symptoms at all. If blood pressure reaches the level of a hypertensive crisis, a person may experience headaches and nosebleeds.

2.6.4 Complications

Long-term hypertension can cause complications through atherosclerosis, where the formation of plaque results in the narrowing of blood vessels. This makes hypertension worse, as the heart must pump harder to deliver blood to the body.

Hypertension-related atherosclerosis can lead to:

- Heart failure and heart attacks;
- Aneurysm, or an abnormal bulge in the wall of an artery that can burst, causing severe bleeding and, in some cases, death;
- Kidney failure;
- Stroke;
- Amputation;
- Hypertensive retinopathies in the eye, which can lead to blindness.

Regular blood pressure testing can help people avoid the more severe complications. However, most people don't even know that they have this sort of disease. In this way, they are not concerned about making the test a habit to prevent serious problems.

2.7 FINAL CONSIDERATIONS

This chapter presented a background about concepts and how researches have been conducted towards e-Health universe.

Several devices were presented, such as AD8232 ECG sensor, arduino and nodeMCU boards. The pattern of send messages in a lightweight way were presented, as well as it's elements. The concept and elements of agents and multi-agent systems were presented, and the environment properties. An explanation about fuzzy logic were presented and the concepts and aspects involving high blood pressure.

3 RELATED WORKS

This chapter describes some works that have relation with the present research. It is divided into the following themes: Publish/Subscribe Architecture for Healthcare, a Fuzzy Logic approach for IoT, Multi-agent system, and wearable devices. This chapter also presents the state of the art considering devices and applications related to e-Health.

3.1 PUBLISH/SUBSCRIBE ARCHITECTURE FOR HEALTHCARE

Many types of research have already pointed out the value of information exchange in health information systems, both regarding the benefit to patients and economic benefits. A study did in 2005 estimated that fully standardized health information exchange can yield as much as U$ 77.8 billion in the United States (CARLIER; RENAULT, 2016). Since then the expanse and the collection of data have increased exponentially, hence increasing the need for information exchange.

According to (WADHWA *et al.*, 2015), data exchange plays a vital role in public healthcare. Having timely access to right data can allow detection of the spread of disease, assess the effectiveness of government schemes, etc. To improve policies and provide awareness, various government healthcare schemes are run which generate a huge amount of data. This data is collected by a number of field workers and is then transferred to central repositories for further analysis. While the data eventually reaches researchers and scientist for analysis, the delay (usually in months) hampers timely actions that can be taken.

Considering that, they propose a Publish/Subscribe based Architecture moderated through a web service that can enable an early exchange of healthcare data among different interested parties, e.g. doctors, researchers, and policymakers. Their architecture handles the privacy and security requirements and provides interoperability support for data exchange. They have also highlighted how the architecture deals with the key challenges involved in public health information exchange.

3.2 FUZZY LOGIC APPROACH FOR IOT

In the last few years, the process of monitoring people health status has grown in interest in the researcher community and led to the development of new devices able to detect and analyze information gathered from many kinds of sensors. These devices are commonly designed to monitor or diagnose disease in the medical field. Moreover, a remarkable interest is growing in the area of sports. In amateur activities such as jogging, running, climbing a set of smart devices are used to improve and monitor performances. Another field of interest is represented by those people that want to monitor their health status by using low-cost devices.

In this way, (SANTAMARíA *et al.*, 2016) proposed a two-stage fuzzy logic approach in which the device tries to learn and fit customer habits to discover outlier warning signals. The two stages approach proposed consists of monitoring the normal activities of the user to build a reference of its condition; Then real-time monitoring and analysis of gathered data from body sensors are accomplished. User status is carried out using a Fuzzy Logic based network. The first stage will give the current activity of the user while the second stage will provide information about health status regarding heart rate.

The achieved results show that after a bootstrap session where the device learns about user habits and its parameters can recognize the base activities making possible to warn the user if something goes wrong about its related heart rate.

3.3 MULTI-AGENT SYSTEM FOR HEALTH

Typically, in MAS, a mobile agent is a piece of autonomous software, capable of migrating between nodes of a network. With IoT in place, we can now have other forms of mobility, being able to move the device physically. Multi-agent systems have been quickly adapted for the e-Health area. Therefore more and more work has been done over the last few years. In a Multi-agent architecture for real-time Big Data processing has been introduced. It has been shown that classical lambda architecture [1] can be extended as a Multi-Agent System.

Toader (2017) proposed a project that will help the medical system gather data in a standard format and the intelligent system will be able to cluster the data to detect various incidents (mass accidents, virus spread, fire incidents, etc.) The presented architecture is scalable, so it will allow us to add or change business logic without any significant restructuring. The chosen technologies are open-source and cross-platform so that they will be easily integrated into most of the existing hardware systems and operating systems.

Periodic analyzer agent will gather data periodically from the sensors and if relevant, will serialize them in the HL7 format (GROUP, 2014) and send them to the cloud. The HL7 standard is a framework for the exchange, integration, sharing, and retrieval of electronic health

[1] Lambda architecture is a data-processing architecture designed to handle massive quantities of data by taking advantage of both batch and stream-processing methods.

information between e-Health software applications. The presented architecture is scalable, so it will allow to add or change business logic without any significant restructuring.

3.4 WEARABLE DEVICES ONTOLOGY

Over the last decade, many technologies have been developed that support individuals in keeping themselves active. This can be done via e-coaching mechanisms and by installing more advanced technologies in their homes. The objective of the Active Healthy Ageing (AHA) Platform is to integrate existing tools, hardware, and software that assist individuals in improving and/or maintaining a healthy lifestyle. This architecture is realized by combining several hardware/software components that generate various types of data. Some examples include heart-rate data, coaching information, in-home activity patterns, mobility patterns, and so on. Multiple subsystems in the AHA platform can share their data in a semantic and interoperable way, through the use of an AHA data-store and a wearable devices ontology.

Considering that Díaz-Rodríguez *et al.* (2017) presented such an ontology for wearable data interoperability in Ambient Assisted Living environments. The ontology includes concepts such as height, weight, locations, activities, activity levels, activity energy expenditure, heart rate, or stress levels, among others. The purpose is serving application development in Ambient Intelligence scenarios ranging from activity monitoring and smart homes to active healthy aging or lifestyle profiling.

3.5 STATE OF THE ART

The growing cost of health care and the aging of population have created many challenges for governments, healthcare providers and healthcare industry that they try to overcome using e-Health technologies, being one emerging area that has potential to improve healthcare service delivery, diagnostic monitoring, disease-tracking and related medical procedures (FELIZARDO *et al.*, 2014).

e-Health solutions are particularly identified by several characteristics, such as context-aware, personalized, anticipatory, adaptive, ubiquity, transparency, performance, security, privacy (ACAMPORA G., 2013). This section presents the state of the art considering e-Health solutions.

3.5.1 ALADDIN project

The ALADDIN project addresses "mild" to "moderate" dementia chronic conditions, and aims to provide an integrated online clinical, educational and social support network for sufferers and their carers in the everyday management of the disease at home (HARITOU M., 2012).

ALADDIN is a platform that provides planning, management and monitoring of the health status. This technology prevents emergency situations due to worsening symptoms, cognitive function, behavioral aspects, in the reduction of stress and burden in the home care situation, and maintaining the patient's and carer's quality of life (CUNO S., 2011). The architecture of ALADDIN-platform consists of three main parts (CUNO S., 2011):

- Carer's Client Application: An application used by carers and patients to access the services of the platform. With this application, carers can answer the questionnaires about the patient's mental health condition, as well as their own. The information about the physiological measurements of the patient can be submitted by the caregiver using the application. Clinicians analyze the provided data on a regular basis. Additionally, the application enables a caregiver to send a warning message to request the clinician to contact the caregiver shortly;
- Server Application: The core of the platform. It implements the basic functionalities of the platform, provides secure communication with client applications, stores information about patients and carers, allows the exchange of information with external Hospital Information Systems (HIS) and provides a web-based graphical user interface for clinicians and platform administrators to interact with the system;
- External Services: Provided by external web portals. There are two types of services involved: cognitive games and a social network. The integration of these services in the platform is achieved by integrating a web browser component in the Carer's Client Application, which opens a web page with the selected external service directly in the client application.

The system helps to ensure that the individual remains as long as possible in its familiar environment. The ALADDIN-platform is designed to stand out by its user-friendliness and simplicity, and it is expected that carer and patient should perceive the technology as a relief (Cuno et al., 2011).

3.5.2 HELP: Home-Based Empowered Living for Parkinson's Disease Patients

Current Parkinson's disease treatments improve symptoms but lead patients to develop a tolerance to the drug. But as the disease progresses, it becomes more difficult to determine the exact drug dosage.

The HELP project aims to improve Parkinson's disease symptoms, by the development of a subcutaneous pump, which injects a gradual and constant flow of apomorphine, a drug derived from morphine, throughout the day, improving the control of symptoms. However, this control method is not sufficient because the pump administers a constant dose, and the appropriate dose is variable and is a consequence of the patient's symptoms that fluctuate throughout the day.

This project has designed a "Parkinson's" sensor which sends relevant information to the platform and automatically establishes the optimal level of drug administration, depending on the state of each patient. At the same time, a constant level of drug is administered by another subsystem developed through the HELP project, consisting of an intraoral device embedded in the patient's mouth in the form of a tooth. This aims to achieve continuous dopaminergic stimulation, improving symptomatic benefits and minimizing concerns about complications resulting from an intermittent dosage of medication (AMBIENT ASSISTED LIVING (AAL), 2012).

The solution proposed by HELP is also highlighted by Telefónica (http://www.tid.es/en/): Telefónica has announced the results of a recently finished pilot that used mobile technology to monitor and treat patients with Parkinson's disease remotely, as the operator and its partners decide on future research and how to commercialize the technology (HANDFORD, 2013).

3.5.3 ROSETTA Project

The ROSETTA project has developed an innovative, integrated system aiming at preventing and managing the problems that can occur as a result of chronic progressive diseases, such as dementia, Parkinson's disease, and Alzheimer. The target group of ROSETTA includes older adults, living in their own houses or small-scale living communities, and their formal and/or informal caregivers.

The ROSETTA system functionalities are as follows:

- The system monitors the activities of the resident employing multiple and different sensors;
- It generates an alarm in case of deviant activity/inactivity or wandering, which is forwarded

to the caregiver (the AAPS part);

- The system generates a warning in the case of long-term variations in the patterns of daily living, which is forwarded to the caregiver (EDS Lifestyle monitoring part);
- It supports the resident directly in carrying out his or her daily activities (EDN Elderly Day Navigator part).

The ROSETTA system is composed of three main parts: Unattended Autonomous Surveillance (UAS-AAPS), Early Detection System (EDS), and Early Day Navigator (EDN) (PROJECT, 2012).

The UAS-AAPS is for surveillance of emergencies, including those as a result of falling and straying, for persons with advanced dementia or persons with a combination of memory problems/mild dementia and Parkinson's disease. The possible emergencies are detected by sensors and a camera and reported to a mobile care team.

The EDS is the lifestyle monitoring system. Sensors in the home monitor the daily life pattern of persons living alone. These regular life patterns are available to care providers as well as informal carers via a computer programme.

The EDN part offers memory support for people with memory problems, through the development of a touch screen and an adapted Smartphone.

3.6 FINAL CONSIDERATIONS

This chapter presents papers that relate to some of the most important aspects of the present research.

(WADHWA *et al.*, 2015) presented a study on an architecture focused on the exchange of information in the health area. Much investment has been made in this field, as this exchange of information plays a vital role in public health. In this way, a Publish / Subscribe architecture was presented, associated with a web service to manage this flow of information.

(SANTAMARíA *et al.*, 2016) presented a study on how the products of monitoring people's vital signs are gaining space in the world market. In the study in question, a network based on fuzzy logic was created, which works in two stages. The first stage captures the current user data and the second stage provides information about the user's heart rate.

Toader (2017) presented a multi-agent system, where a stand-alone mobile agent will capture user information and, if relevant information, will be sent to another agent in the cloud, using the HL7 standard. HL7 is already widespread in health-related software. The architecture is also interesting for its scalability.

Díaz-Rodríguez *et al.* (2017) presented an ontology for an ambient assisted living, involving several concepts and increasing the monitoring of activities in smart homes to improve the lifestyle of its users.

Finally were presented three examples to illustrate the state of the art encompassing e-Health solutions.

Below are presented table 1 and table 2 that consolidates mains information about the related works and state of the art.

Table 1 – Comparison between related works

Related work	Proposal	Field	Details
Publish/subscribe architecture for healthcare	Publish/subscribe based architecture moderated through a web service.	Communication	Exchange of healthcare data among diferente interested parties, e.g. Doctors, researchers, and policymakers.
Fuzzy logic approach for IoT	Two-stage fuzzy logic approach to learn and fit customer habits.	Artificial Intelligence	Monitoring the normal activities of the user to build a reference of its condition and discover outlier warning signals.
Multi-agent system for health	Architecture to help the medical system gather data in a standard format.	Artificial Intelligence	The intelligent system will be able to cluster the data to detect various incidents (mass accidents, virus spread, fire incidents, etc.)
Wearable devices ontology	An ontology for wearable data interoperability in ambient assisted living environments.	Software Engineering	Serve application development in ambient intelligence scenarios ranging from activity monitoring and smart homes to active healthy aging or lifestyle profiling.

Source – Elaborated by the author

Table 2 – Comparison between state of the art projects

State of the art Project	Proposal	Modules / Devices	Details
Aladdin	Planning, management and monitoring of the health status.	Carer's client application; Server application; External services.	Addresses "mild" to "moderate" dementia chronic conditions; Provide an integrated online clinical, educational and social support network for sufferers and their carers in the everyday management of the disease at home.
Help	Improve parkinson's disease symptoms.	Subcutaneous pump; Intraoral device embedded in the patient's mouth in the form of a tooth.	Use a sensor which sends relevant information to the platform and automatically establishes the optimal level of drug administration.
Rosetta	Preventing and managing the problems that can occur as a result of chronic progressive diseases.	Unattended autonomous Surveillance (uas-aaps); Early detection system (eds); Early day navigator (edn).	The system monitors the activities of the resident employing multiple and different sensors; It generates an alarm in case of deviant activity/inactivity or wandering, which is forwarded to the caregiver (the aaps part); The system generates a warning in the case of long-term variations in the patterns of daily living, which is forwarded to the caregiver (eds lifestyle monitoring part); It supports the resident directly in carrying out his or her daily activities.

Source – Elaborated by the author

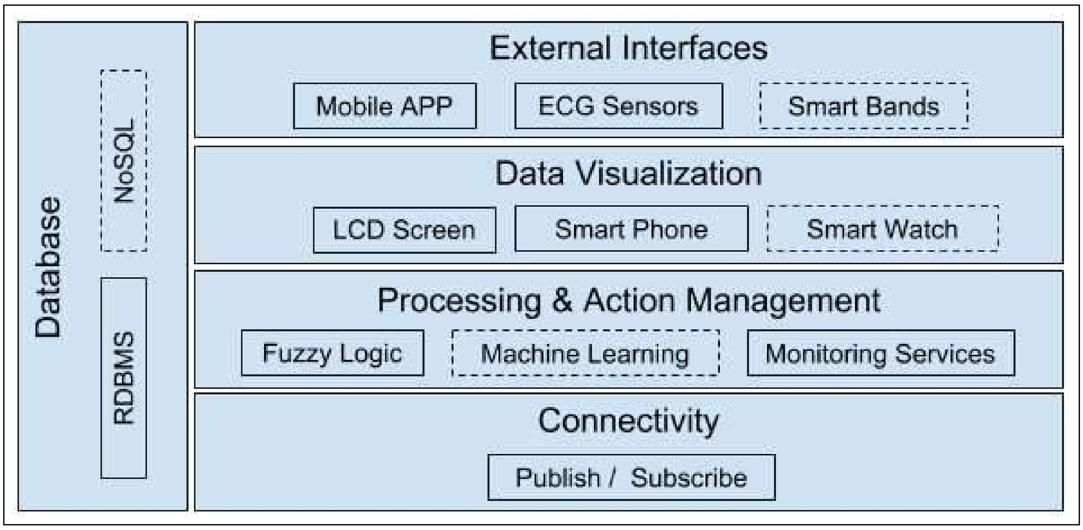
4 A MULTI-AGENT SYSTEM USING FUZZY LOGIC APPLIED TO E-HEALTH TO MONITOR HYPERTENSION

This chapter presents the proposal of the whole environment, including architecture and other elements of MAS.

4.1 ELEMENTS OF THE ARCHITECTURE

The proposed architecture consists of five main elements - External Interfaces, Data Visualization, Processing & Action Management, Connectivity, and Database - as shown in Figure 13, and described below. The elements with dotted lines have not yet been implemented and will be presented as a proposal for future work.

Figure 13 – Proposed architecture.



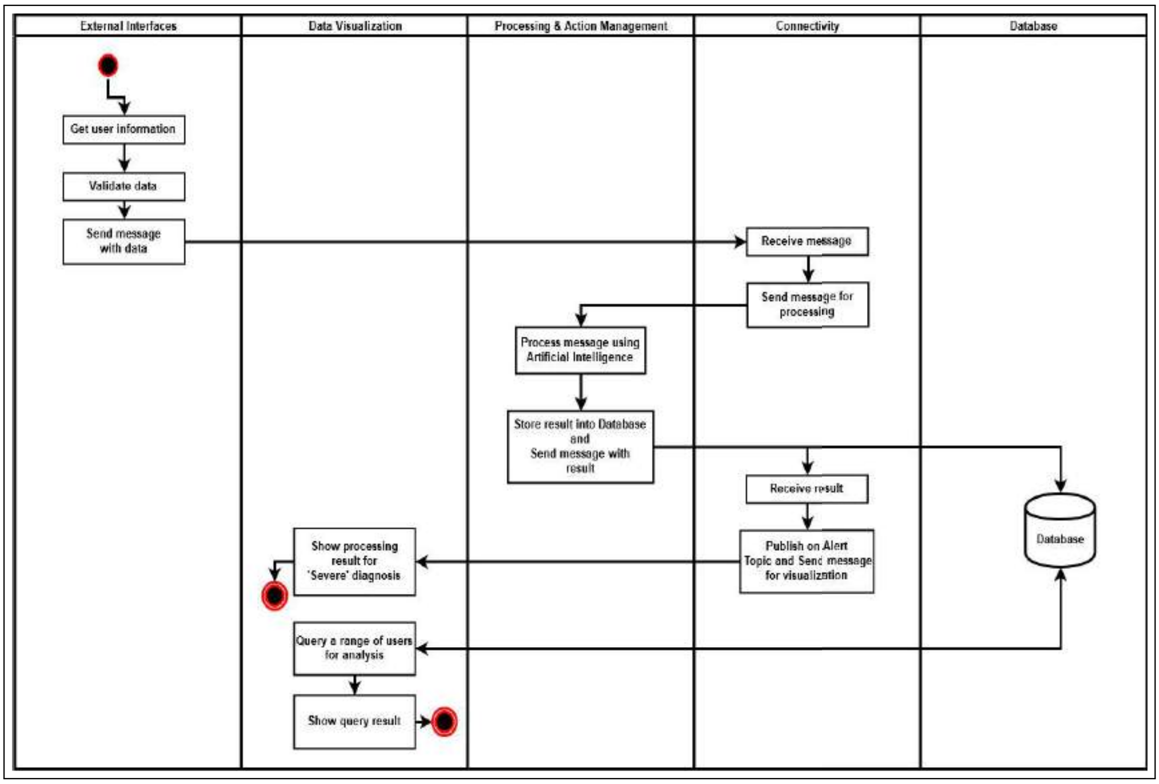
Source – Elaborated by the author

Figure 14 shows how the elements interact with each other. The details of that interaction are explained in the subsections below.

4.1.1 External Interfaces

This section explains how the MAS will perceive the world through devices with sensors. The user will interact with the solution through devices that allow insert the data manually or collect them automatically. This point is the only entrance for the data. Moreover, the data must be validated to assure the invalid data won't be sent to the next element of the architecture.

Figure 14 – Interaction between elements of the architecture



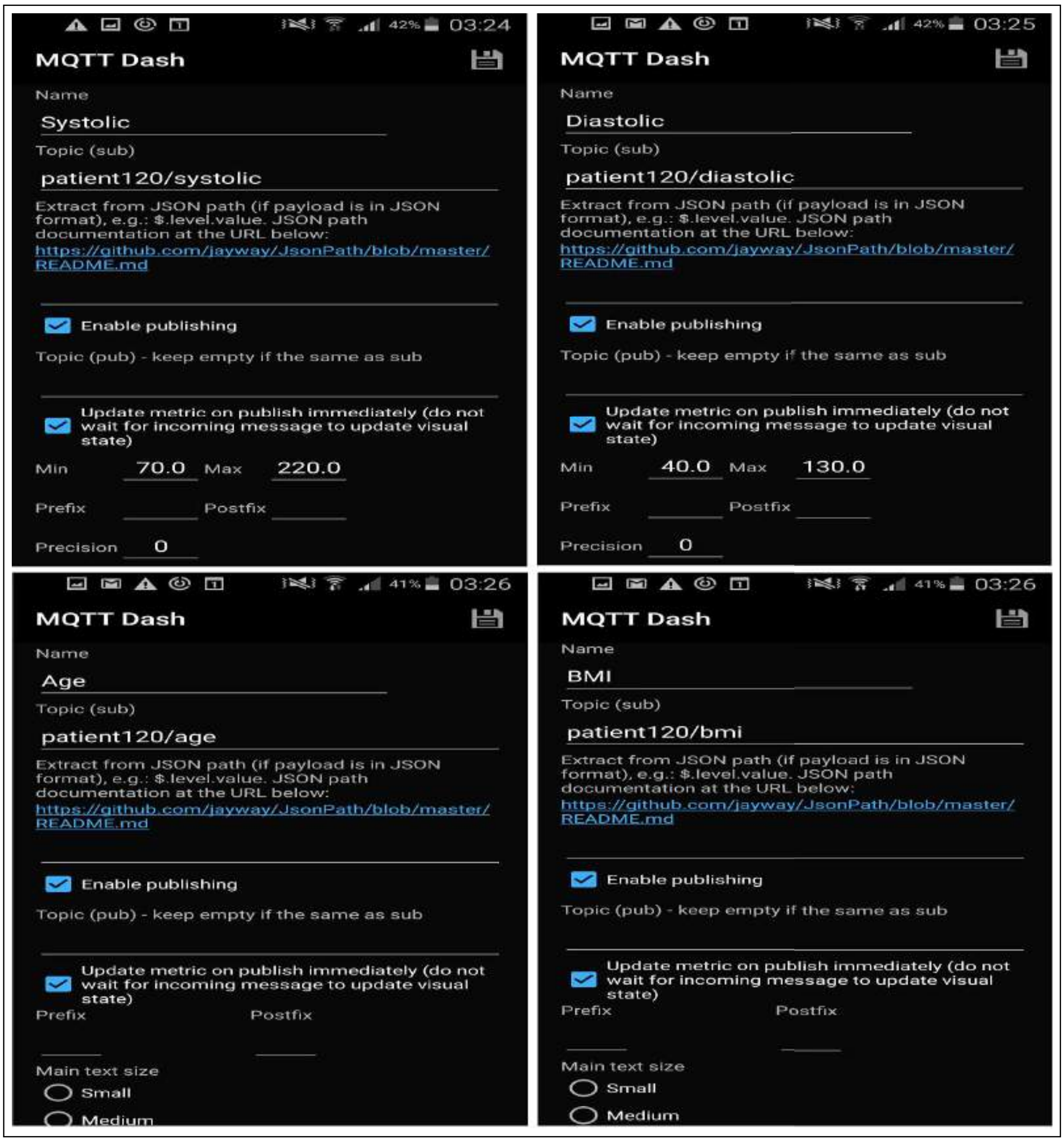
Source – Elaborated by the author

4.1.1.1 Mobile APP

Mobile APP-related to e-Health is becoming popular over time, because they can easily substitute paper and spreadsheets, and validate data before sending to the server. Considering this proposed MAS solution, an APP representing one element of the architecture allows set up input fields and verify them before submitting the data to be used, as shown in Figure 15.

The screenshot shows the mobile application screens for configuring user input fields. The image of the first quadrant refers to systolic blood pressure. It is possible to configure the topic in MQTT to publish the inserted information (TOPIC (SUB)). In addition, the user can set the minimum and maximum values to validate the inserted data. The other fields can be filled with the default option. The image of the second quadrant refers to the diastolic blood pressure. It is possible to configure the topic in MQTT to publish the inserted information (TOPIC (SUB)). Likewise, the user can also set minimum and maximum values to validate the entered data. The other fields can be filled with the default option. The image of the third quadrant is related to age. The user can configure the topic in MQTT to publish the inserted information (TOPIC (SUB)). The other fields can be filled with the default option. The image of the fourth quadrant is related to BMI. The user can configure the topic in MQTT to publish the inserted information (TOPIC

Figure 15 – Get Patients' Data



Source – Elaborated by the author

(SUB)). The other fields can be filled with the default option.

Instead of register information on paper spreadsheets, running the APP in smartphones makes easy to the user to input the data of systolic, diastolic, age and BMI, as shown in Figure 16, and helps the user to input data correctly.

4.1.1.2 ECG Sensors

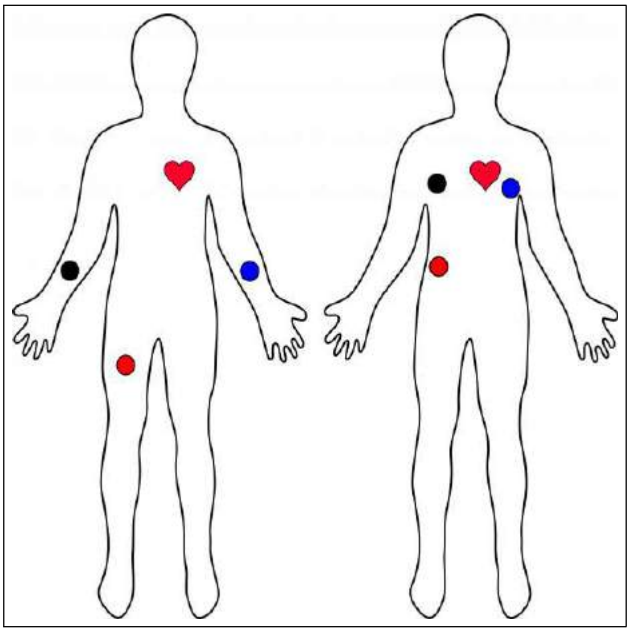
An ECG (electrocardiogram) captures the electrical activity of a heart at rest. It provides information about heart rate and rhythm and shows if there is enlargement of the heart due to high blood pressure (hypertension) or evidence of a previous heart attack (myocardial infarction). The sensors can be placed in different parts of the body, considering the way of a

Figure 16 – Mobile APP Running

Source – Elaborated by the author

measure of the signals, as shown in Figure 17.

Figure 17 – Typical Sensor Placements

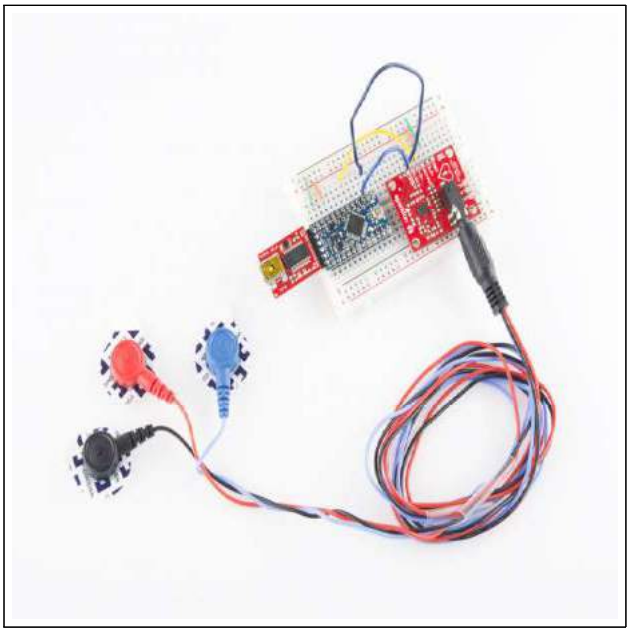


Source – sparkfun.com

An AD8232 shield and nodeMCU board were used to implement the proposed

MAS/IoT solution. Regarding estimation of blood pressure using ECG signals, was followed by Kumar et al. (KUMAR; AYUB, 2015). Figure 18 depicts the sensors coupled on Arduino board, which send the signals through Wi-Fi.

Figure 18 – Sensors connected to Heart Monitor



Source – sparkfun.com

The following part of code that get signals from ECG sensors is presented on Source 1. It represents an example of a mobile agent of the proposed architecture. The full code is available on Appendix B.

Source Code 1 – CATCHING ECG SIGNAL

```
... # Some processing is made here
  Serial.begin(9600);
  pinMode(10, INPUT); // Setup for leads off detection LO +
  pinMode(11, INPUT); // Setup for leads off detection LO -
  if((digitalRead(10) == 1)||(digitalRead(11) == 1)){
    Serial.println("!");
  }
  else{
    // send the value of analog input 0:
      Serial.println(analogRead(A0));

```

```
... # The processing continues...
```

The digital ports 10 and 11 of the nodeMCU receives information from AD8232 board and detects when a treshold (value = 1) is reached. This case reflects bad signal that must be ignored. Otherwise, the value read from analog port 0 is sent to the serial port and will be analized.

4.1.1.3 Smart Bands

Just a couple years ago, fitness bracelets were basically glorified step counters worn on user's wrist. Now, they're doing everything from measuring heart rate on a run to warning the user to get out of the sun. And they're everywhere. Considering the convenience and practicality in capturing vital signs without the intervention of the user and at different times, the use of smart bands that can communicate via Bluetooth is an exciting and viable option. Moreover, the data are validated automatically, avoiding mistakes when these data are processed and analized. We intend to let the architecture ready to receive information from these devices in the close future.

4.1.2 Data Visualization

This section explains how the MAS will show information to the user. According to Figure 14, when the processing of information results in 'Severe' diagnosis, the solution will show the result and warning message to the involved user, letting him take some action as soon as possible. Besides, this element will show the result of specific queries on the database abaout the information from one or more users made by authorized people.

4.1.2.1 LCD Screen

Depending on where the solution will work, the information resulting from the processing can be viewed on a Liquid Crystal Display (LCD) screen, where the allowed users can perform several types of queries varying the filters. Considering a hospital scenario, a doctor can access information from a specific user through the screen os his laptop. In addition, nurses can check on almost realtime information from patients on a TV screen installed on nurse's sector, without the necessity to get the data manually.

4.1.2.2 Smartphone

With the advent of mobile applications, there aren't obstacles for the user to access internet and take any information he wants. In this scenario, the user can see the result of processing his information, regarding its vital signals, using any smartphone, as long as he has granted access to that query, and check on almost realtime how his health is going.

4.1.2.3 Smartwatch

Just as with user's smartphone, internet access enables a smart watch with a whole world of potential capabilities, like message notifications, GPS navigation and calendar synchronization. And of course, a Bluetooth connection to the phone means the watch can help the user place calls or send and receive messages. Considering this, the user can receive messagens with the result of processing, regarding its vital signals, using any smartwatch. We intend to let the architecture ready to send information to these devices.

4.1.3 Processing & Action Management

This section explains, as shown on Figure 14, how the MAS processes the user's data, using artificial intelligence approach, learning automatically over time to suggest new approaches about the health status of the user, and monitoring core services involving the inference of the level of high blood pressure. Moreover, the result information will be sent to a database for storing as well as send the result for users with 'Severe' diagnosis. In addition, this element has a internal service just for monitor the others services involved to garantee they will be always available.

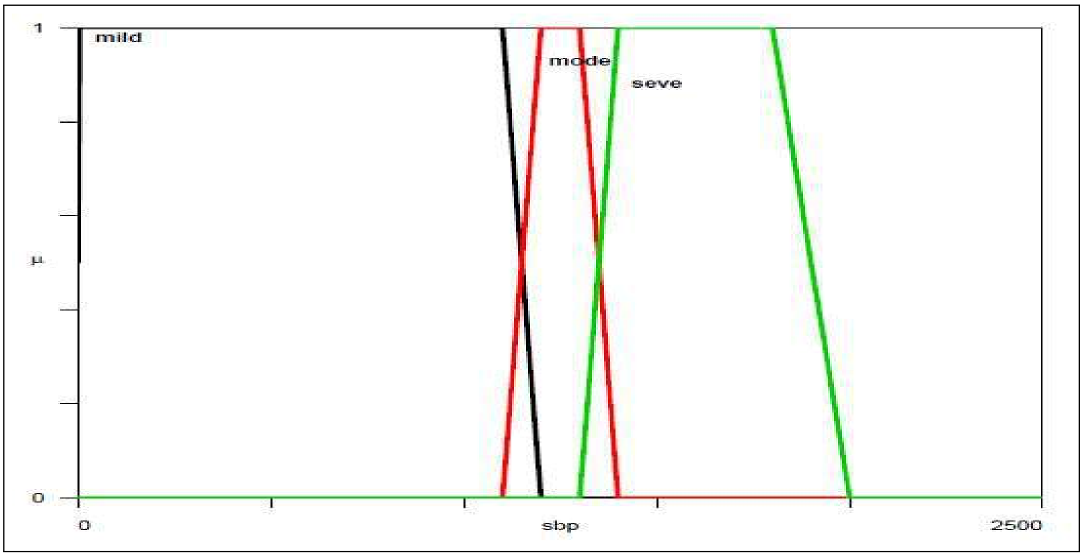
4.1.3.1 Fuzzy Logic

The Fuzzy Logic was chosen considering getting a better performance regarding almost real-time inference of blood pressure level. In Chandra (2014) it is proposed a web-based fuzzy expert system for the management of hypertension (High Blood Pressure), whose classifiers can detect the hypertension risk as Mild, Moderate or Severe.

Supported by the experts, some adjustments were made on the classifiers to keep them aligned to 7th Brazilian Director of Blood Hypertension (MALACHIAS, 2016). Figures 19 to 22 show the membership functions related to each input. The input membership functions considered the following:

1. Systolic Blood Pressure (SBP) (Trapezoidal): Mild, Moderate, Severe;

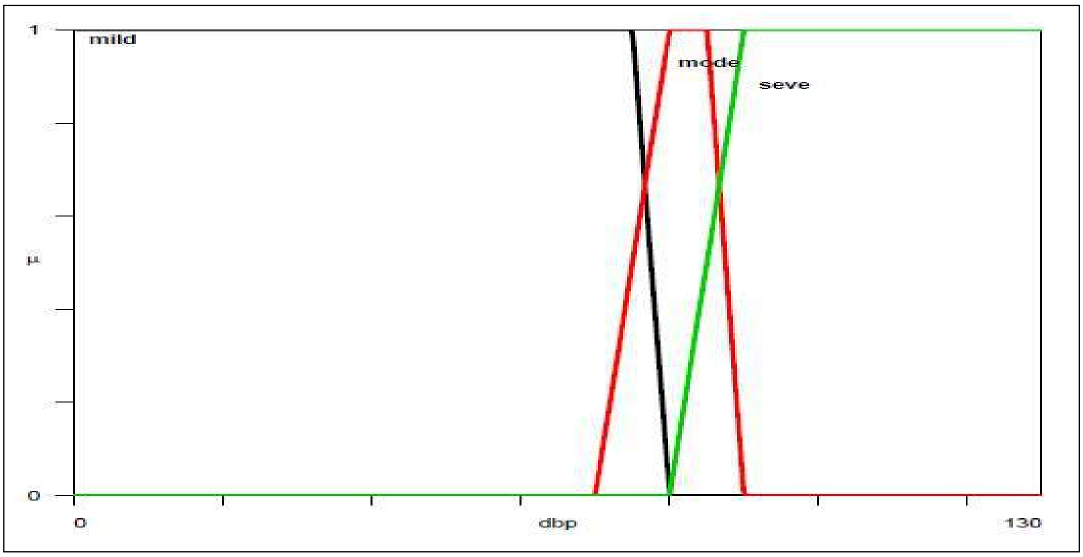
Figure 19 – Systolic Blood Pressure



Source – Elaborated by the author

2. Diastolic Blood Pressure (DBP) (Trapezoidal): Mild, Moderate, Severe;

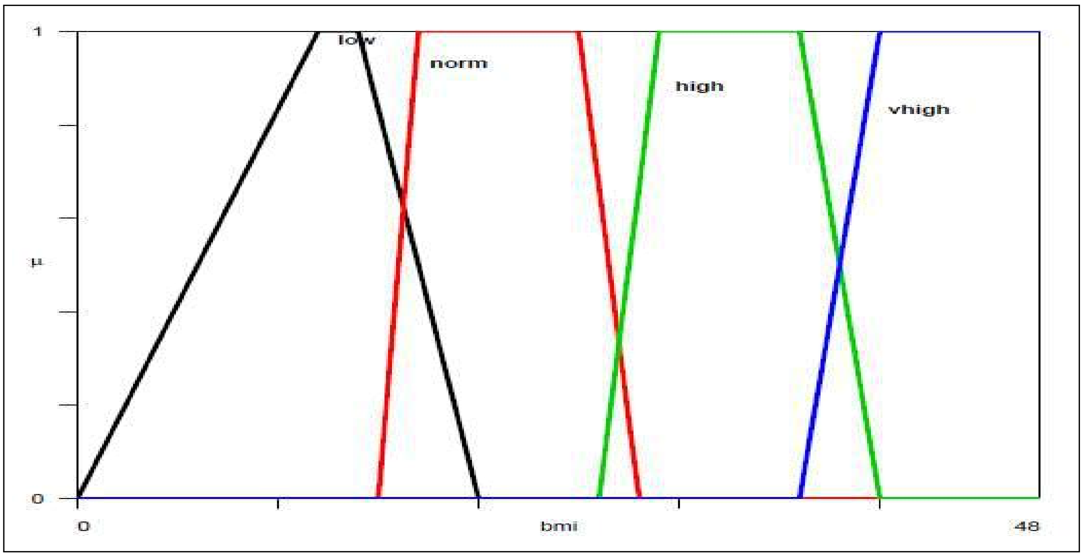
Figure 20 – Diastolic Blood Pressure



Source – Elaborated by the author

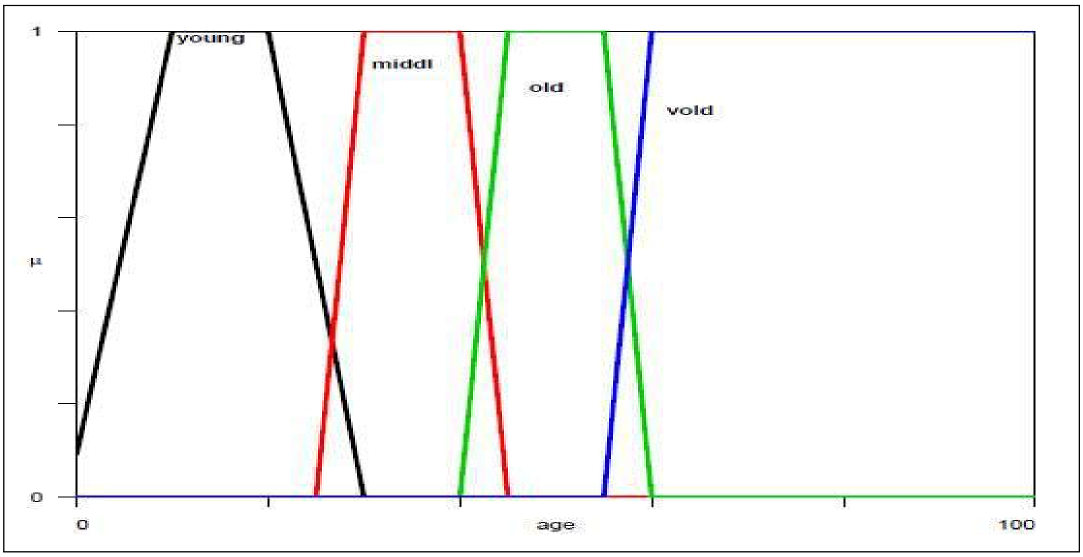
3. Age (Trapezoidal): Young, Middle-Aged, Old, Very Old;
4. Body Mass Index (Trapezoidal): Low, Normal, High, Very High.

Figure 21 – Body Mass Index



Source – Elaborated by the author

Figure 22 – Age



Source – Elaborated by the author

The output membership function considered the following, as shown in Figure 23:

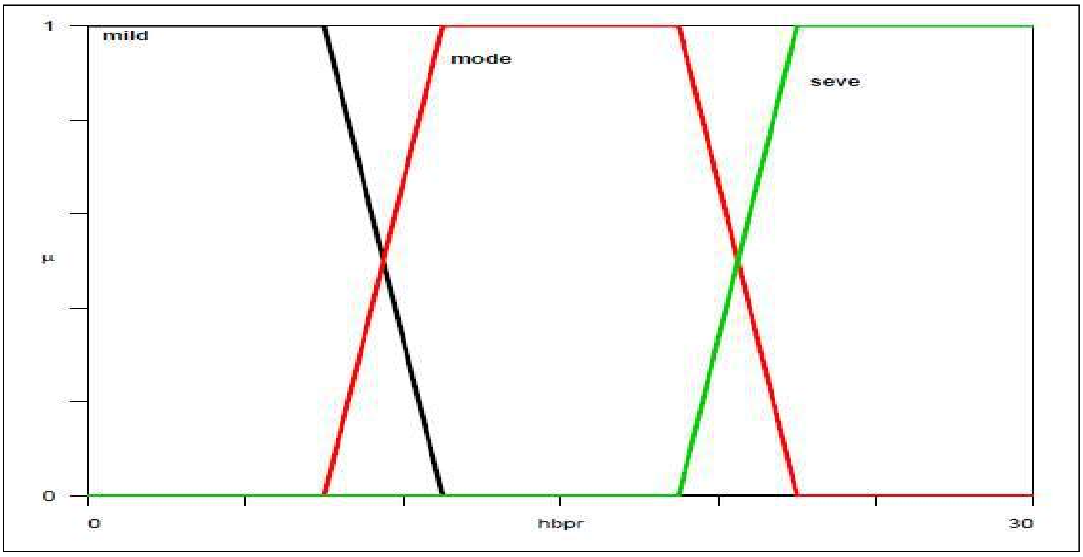
1. Blood Pressure (Trapezoidal): Mild, Moderate, Severe.

The configuration steps of the fuzzy inference system generated 108 rules that were used by the processing agent to infer the hypertension level.

In R software, the fuzzy inference system was set up using the following parameters:

1. Fuzzy Inference System = Mamdani;
2. AND Method = Min;
3. OR Method = Max;

Figure 23 – Output Membership Function



Source – Elaborated by the author

4. Implication Method = Min;
5. Aggregation Method = Max;
6. Defuzzification Method = Centroid, Largest of Maximum and Middle of Maximum.

Despite the fact that a Sugeno system is a more compact and computationally efficient representation than a Mamdani system, we used the available library in R software that has only Mandani Inference System implemented so far.

There are several defuzzification methods available. Due to the limitations of the library in R software, we choose only the Centroid, Largest of Maximum and Middle of Maximum methods to implement our Fuzzy logic.

4.1.3.2 Machine Learning

As time goes by, it is essential that the system begins to understand better what can affect the increase in user pressure level, taking into account both the user's data and comparing with other users with similar characteristics. We intend to use machine learning techniques, to help both the user and health professionals who will accompany a specific group of users, thus allowing a faster, more efficient and more precise action.

4.1.3.3 Monitoring Services

From the moment that fuzzy logic infers a diagnosis of blood pressure level as severe, it is essential that the system be prepared to act and alert promptly both the user and the team that accompanies it, when applicable. The Source 2 shows a code snippet where the system sends a warning signal to the user who had pressure level classified as severe. The full code is available on Appendix E.

Source Code 2 – PUBLISH ON ALERT TOPIC

```
# ... Some processing is made here
# Reading rows
for row in cursor:
    result_print = "Alert to Patient: "+str(row[1])+ ";
       Result: "+ str(row[5])
    client.publish("result_"+str(row[1]), result_print, qos
       =1)
cursor.close()
# The processing continues...
```

4.1.4 Connectivity

This section explains how the MAS will use techniques that improve fast data exchange to connect the whole pieces of the solution in an asynchronous environment.

4.1.4.1 Publish / Subscribe

The Publish/Subscribe model allows messages to be broadcast to different parts of a system asynchronously. A sibling to a message queue, a message topic provides a lightweight mechanism to broadcast asynchronous event notifications and endpoints that allow software components to connect to the topic to send and receive those messages.

This MAS used Message Queuing Telemetry Transport (MQTT) protocol, which is an implementation of Publish/Subscribe model and provides a lightweight and flexible network protocol that offers the ideal balance for IoT devices involved into the proposed architecture.

4.1.5 Database

This section explains how the MAS will store the raw and processed information. There are two main paradigms related to database, which are described below. Both should be considered in this solution, because each one has its own features and applications.

4.1.5.1 RDBMS

A Relational Database Management System (RDBMS) is a set of formally described tables from which data can be accessed or reassembled in many different ways without having to reorganize the database tables. The standard user and Application Programming Interface (API) of a relational database is the Structured Query Language (SQL). SQL statements are used both for interactive queries for information from a relational database and for gathering data for reports.

Elements of the relational database management system that overarch the basic relational database are so intrinsic to operations that it is hard to dissociate the two in practice. The most basic RDBMS functions are related to create, read, update and delete operations, collectively known as CRUD. They form the foundation of a well-organized system that promotes consistent treatment of data.

Each table, which is sometimes called a relation, in a relational database contains one or more data categories in columns, also called attributes. Each row, also called a record or tuple, contains a unique instance of data, or key, for the categories defined by the columns. Each table has a unique primary key, which identifies the information in a table. The relationship between tables can then be set via the use of foreign keys - a field in a table that links to the primary key of another table.

RDBMS uses complex algorithms that support multiple concurrent user access to the database, while maintaining data integrity. Security management, which enforces policy-based access, is yet another overlay service that the RDBMS provides for the basic database as it is used in enterprise settings.

A RDBMS was chosen considering the necessity to save the data read from sensors and mobile APP, identify them for processing and allow queries for specific purposes. The stored information should be precisely retrieved and updated after processing and diagnosis inference. The implementation was made using MySQL, as shown on Source 3. The full code is available on Appendix C.

Source Code 3 – RDBMS Operations

```

# ... Some processing is made here

# Reading information from MQTT and store into MySQL
    receivedmsg = str(message.payload)
    patient=patient[patient.index("=")+1:]
    sistolic=sistolic[sistolic.index("=")+1:]
    sistolic_aux=int(sistolic)*10
    sistolic=str(sistolic_aux)
    diastolic=diastolic[diastolic.index("=")+1:]
    bmi=bmi[bmi.index("=")+1:]
    age=age[age.index("=")+1:]
    dateinsert=dateinsert[dateinsert.index("=")+1:]
    comando = " INSERT INTO input2process (V1, V2, V3, V4,
       V5, V6, V7) " + \
                   " VALUES " + \
                "( "+patient+" , "+sistolic+" , "+diastolic
                   +" ,"+bmi+", "+ age + \
                "  , WaitingProc , "+dateinsert+" )"
    print(comando)
    cursor.execute(comando)
    db.commit()

# The processing continues...
```

The data is received from the broker (MQTT), and splited for each respective field. Then, organized data is stored on the database.

4.1.5.2 NoSQL

not only SQL (NoSQL) represents an entirely different framework of databases that allows for high-performance, quick processing of information at a massive scale. NoSQL is an alternative to traditional relational databases in which data is placed in tables and data schema is carefully designed before the database is built. In other words, it is a database infrastructure that is very well-adapted to the heavy demands of big data.

Early NoSQL databases for web and cloud applications tended to focus on very specific characteristics of data management. The ability to process very large volumes of data and quickly distribute that data across computing clusters were desirable traits in web and cloud design. Developers who implemented cloud and web systems also looked to create flexible data schema – or no schema at all – to better enable fast changes to applications that were continually updated.

We intend to implement interaction with NoSQL database to speed up access to significant data and allow the use of machine learning techniques without compromising system performance.

4.2 MAS SOLUTION ARCHITECTURE USING IOT

As stated by Russel and Norving (RUSSELL; NORVIG, 2010), the task environment involves the PEAS (Performance, Environment, Actuators, Sensors) that should be described when designing an agent. The architecture proposed consists of three agents, as described in table 3.

Table 3 – PEAS Description of the Task Environment

Agent Type	Performance Measure	Environment	Actuators	Sensors
Mobile	Correct Signals	Home	Pub/Sub Topic	ECG sensors / Mobile APP
Processing	Diagnosis Accuracy	Cloud	Pub/Sub Topic	Pub/Sub Topic
Monitoring	Diagnosis Availability Time	Monitor centre	Mobile Phone Screen / Monitor Screen	Pub/Sub Topic

Source – Elaborated by the author

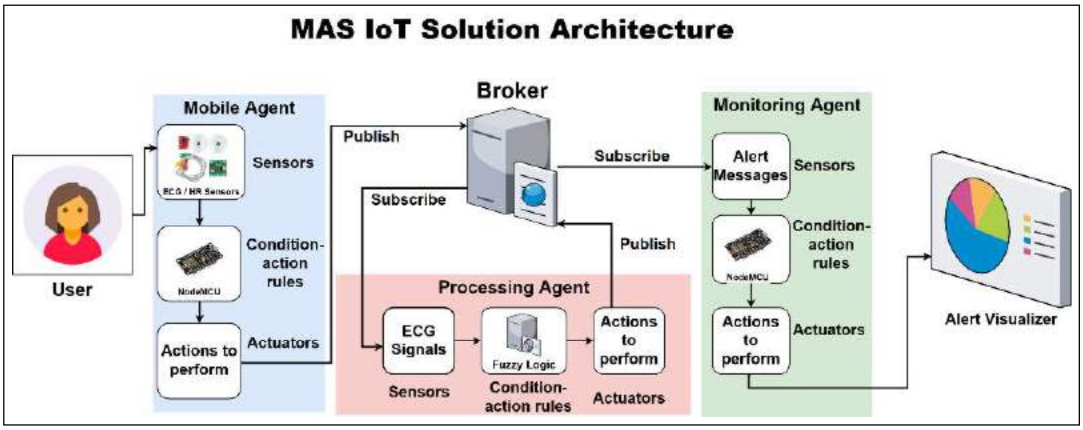
The characteristics of the task environment of the three agents are:

deterministic - episodic - ; static - and discrete -

- Accessible - there is complete information about the environment;
- Deterministic - the rules used by the agent has specific results;
- Episodic - each processing task is independent from another;
- Static - only the agents can modify the environment;
- Discrete - there is a finite rules to be used by the agent;
- Multi-Agent - three agents involved.

The embedded multi-agent architecture that we propose for our MAS/IoT solution using Publish/Subscribe Pattern is resumed in Figure 24.

Figure 24 – Architecture proposed - MAS (3 agents).



Source – Elaborated by the author

All Agents works separately. The Mobile Agent works catching the vital signs of the user. The Processing Agent can process information through fuzzy logic and infer how the blood pressure of the user is going, based on four parameters - systolic blood pressure, diastolic blood pressure, age, and body mass index, and indicates some abnormality. The Monitoring Agent gets the process result which is shown on an output device. Publish/Subscribe Pattern allows the connectivity of this environment. This figure is another perspective of the Figure 14, fully described in section 4.1.

Considering the agents of the solution, the agent program of the mobile agent, as shown in Algorithm 2, perceives the vital signals (systolic and diastolic blood pressure), age and body mass index of the user/patient.

If some constraints aren't respected, such as diastolic equals to zero, the agent program cancels the processing and waits for another cycle of running. Otherwise, the information are sent to the MQTT Broker.

The agent program of the processing agent, as shown in Algorithm 3, perceives that there are information published on the broker and get those data.

The data is stored in the database and the fuzzy inference system process the information and infers the diagnosis for each user/patient, considering the rules defined in Appendix A. The result is stored in the database and if its value equals to "severe," it is published on MQTT Broker. The code for getting data from MQTT and store in the database is available on Appendix C. The code of fuzzy inference system is available on Appendix D.

Algorithm 2: Mobile Agent Program

Input: percept

Output: an action

persistent: rules, a set of condition–action rules
state←INTERPRET-INPUT(percept)
rule←RULE-MATCH(state, rules)
if(state.systolic between X and Y) and
(state.diastolic between A and B) and
(state.age between F and G) and
(state.BMI between Z and W) then send_data
else CANCEL
action ←rule.ACTION
return *action*

Algorithm 3: Processing Agent

Input: percept

Output: an action

persistent: rules, a set of condition–action rules
state←INTERPRET-INPUT(percept)
rule←RULE-MATCH(state, rules)
if (SBP is severe AND DBP is severe) then severe
elseif (SBP is mild AND DBP is moderate AND Age is (young OR
middle-aged) AND BMI is (HIGH OR Very high))
comment - other rules are related to "moderate" level
then moderate
comment - other rules are related to "mild" level
else mild
action ←rule.ACTION
return *action*

The agent program of the monitoring agent, as shown in Algorithm 4, perceives that there is information published on the broker and get those data, sending information to the remote device in order to alert the user/patient. The monitoring agent can show the information on devices, such as monitor screen or tablet, according to filters defined by granted users. The code for get data from the database and publish on MQTT is available in Appendix E.

Algorithm 4: Monitoring Agent

Input: percept

Output: an action

persistent: rules, a set of condition–action rules
state←INTERPRET-INPUT(percept)
rule←RULE-MATCH(state, rules)
if (state.alert = 1) **then** start_device
else stop_device
action ←rule.ACTION
return *action*

The code of all services and applications are available on GitHub [1].

[1] https://github.com/afonsoblneto/eHealth

5 EVALUATION AND RESULTS

This chapter presents the assessment of the proposed architecture regarding the application of the fuzzy logic to infer the level of blood pressure in two scenarios: using all agents for just one patient, and using processing and monitoring agents for several patients obtained from a public database.

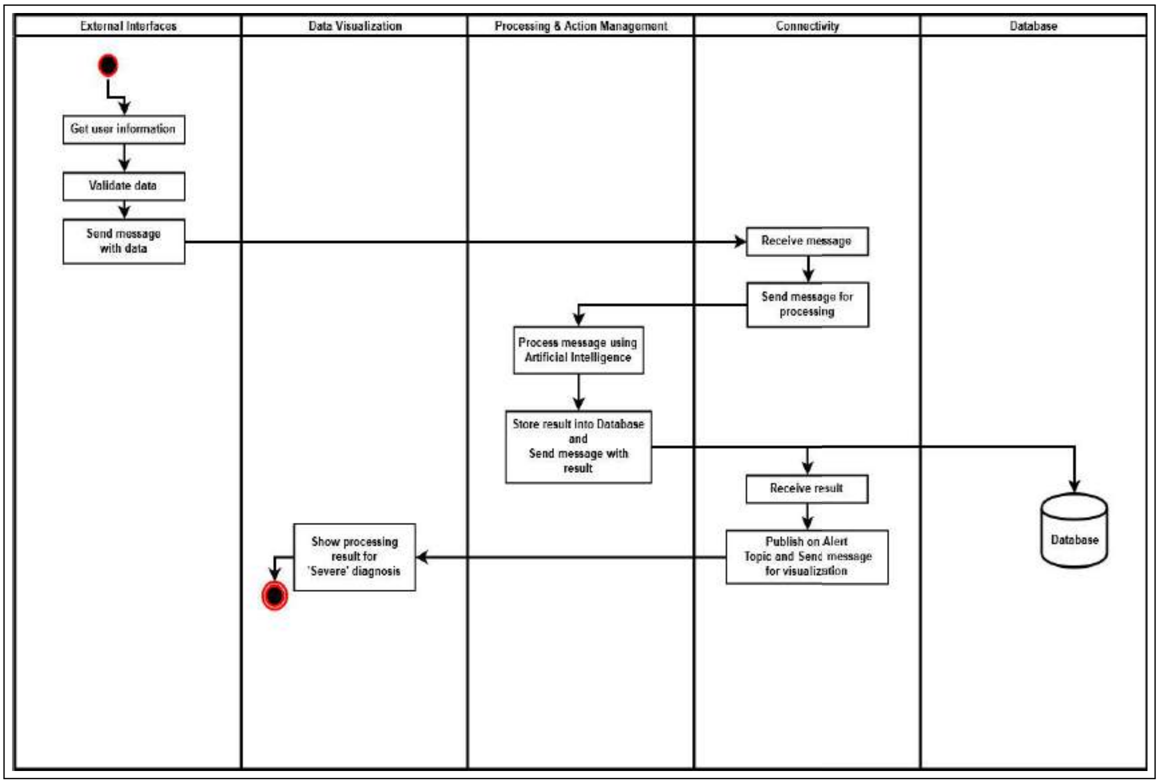
5.1 EVALUATION

The solution has been evaluated in two ways, to test all agents involved, and test the accuracy and performance of fuzzy inference system, using real data from 768 distinct patients from "Pima Indians Diabetes Data Set" from UCI Machine Learning Repository Database (SIGILLITO, 2013).

5.1.1 Using all Agents to test the whole solution

Evaluating the scenario where the three agents are available, we used the proposed MAS solution considering the following, as shown in Figure 25 and descripted below:

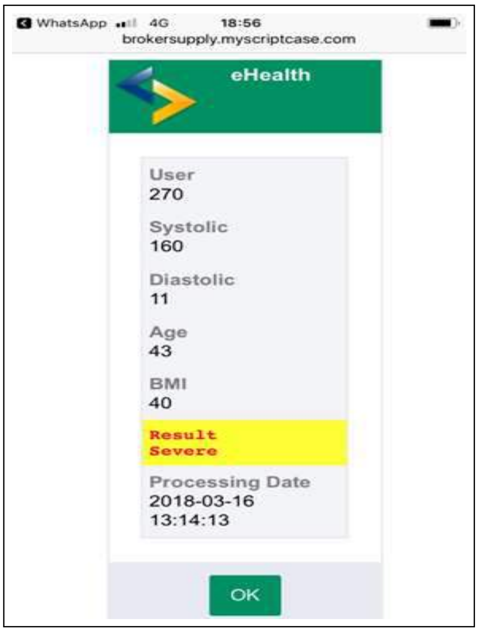
Figure 25 – Using the three agents of the solution



Source – Elaborated by the author

1. Mobile Agent (External Interfaces): consists in a mobile MQTT client app used to get the user's systolic and diastolic pressure, body mass index, and age, validate them and publish on a topic named "measurements" on MQTT broker;
2. Publish/Subscribe (Connectivity) pattern implemented using MQTT broker (MQTT, 2018);
3. MySQL (Database) to store the information of the user;
4. Processing Agent (Processing & Action Management): There is a subscriber service developed in python that gets those data on MQTT and saves them in a MySQL database. As proposed in (CHANDRA, 2014), we developed a fuzzy expert system using R (R-PROJECT, 2018), for the management of hypertension (High Blood Pressure), that classifies the hypertension risk as Mild, Moderate or Severe. The results are stored in the database and published on a topic named "alerts" on MQTT broker when the diagnosis is "Severe." To test the agent, the "severe" diagnosis was intentionally reported;
5. Monitoring Agent (Data Visualization): The result was presented on a smartphone screen, as shown in Figure 26.

Figure 26 – Result of User Data Processing



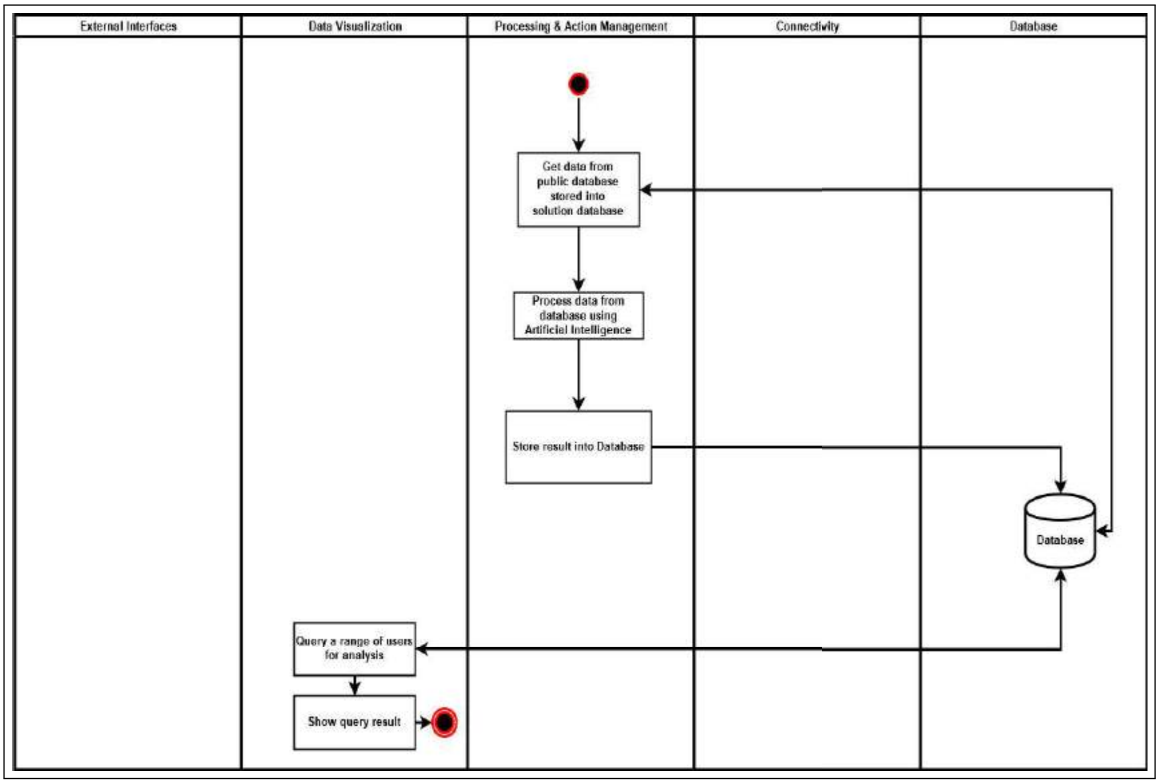
Source – Elaborated by the author

The result shown on APP about the user ID = 270, where based on systolic, diastolic, age and BMI, the MAS inferred that the level of blood pressure is "Severe," suggests that the user take some action as soon as possible, such as a call to a doctor.

5.1.2 Using public data from 768 patients to test the accuracy and performance of the Fuzzy logic

Evaluating the scenario where the Fuzzy logic accuracy and the performance are tested, we used the proposed MAS solution considering the following, as shown in Figure 27 and descripted below:

Figure 27 – Testing performance and accuracy of the Fuzzy logic



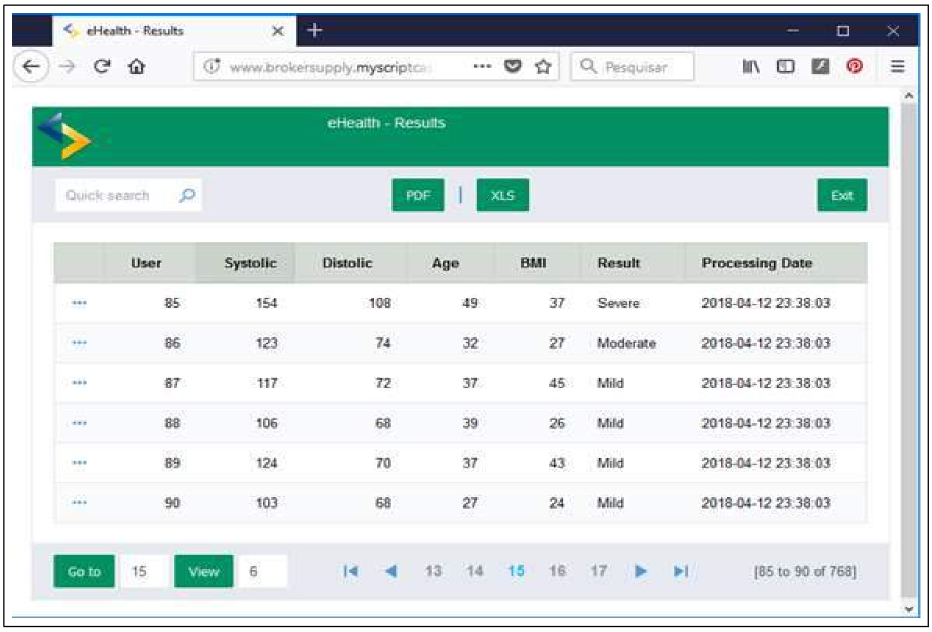
Source – Elaborated by the author

1. No External Interfaces were used;
2. Publish/Subscribe (Connectivity) pattern implemented using MQTT broker (MQTT, 2018);
3. MySQL (Database) using real data from 768 distinct patients from "Pima Indians Diabetes Data Set" from UCI Machine Learning Repository Database (SIGILLITO, 2013);
4. The defuzzification methods used were Centroid, Middle of Maximum and Largest of Maximum (Processing & Action Management);
5. Processing Agent (Processing & Action Management): There is a subscriber

service developed in python that gets those data on MQTT and save them in a MySQL database. As proposed in (CHANDRA, 2014), we developed a fuzzy expert system using R (R-PROJECT, 2018), for the management of hypertension (High Blood Pressure), that classifies the hypertension risk as Mild, Moderate or Severe. The results are stored on the database;

6. Monitoring Agent (Data Visualization): The results were presented on a website that access the database and shows the results according to available filters (see Figure 28).

Figure 28 – Inference of Level of Blood Pressure for Several Patients



Source – Elaborated by the author

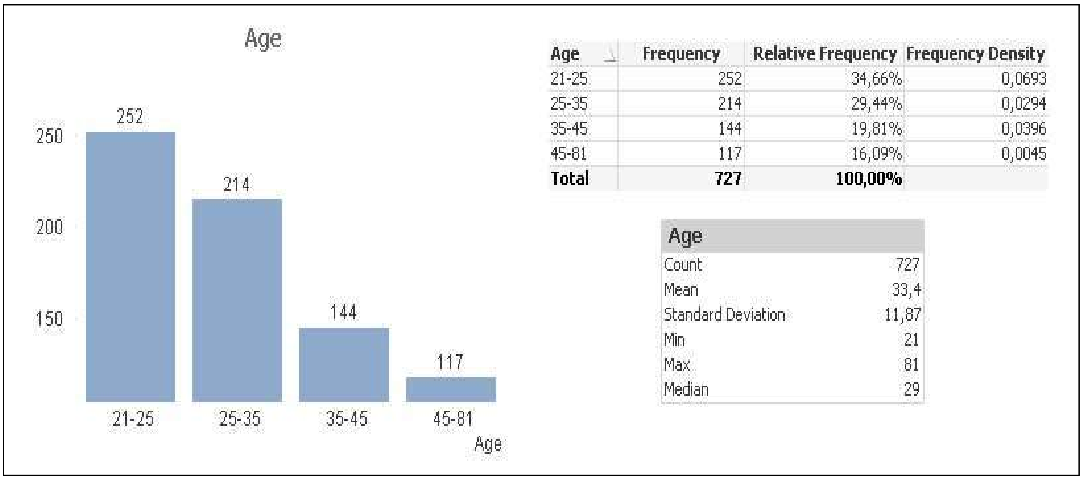
The website shows information (systolic, diastolic, age, BMI) from several patients, as well as the result of the inference made by the MAS for each patient. This website has restricted access only to authorized users.

5.2 RESULTS

Using all agents was possible to check that all functionalities worked fine. Each agent performed its role and showed the result of inference by Fuzzy Logic on the smartphone. During the tests using all agents, some inconsistencies were detected when getting ECG signals to calculate systolic and diastolic blood pressure, indicating the necessity of improvements in the code. Thus, the mobile app was the only one used during the tests representing the mobile agent.

Considering the accuracy of Fuzzy Logic and focusing in processing agent, the patients' data were available according to Figures 29 to 32, where each parameter was split into classes, and their respective frequencies, relative frequencies, and frequency densities were calculated making more straightforward to understand the data distribution.

Figure 29 – Age statistical information



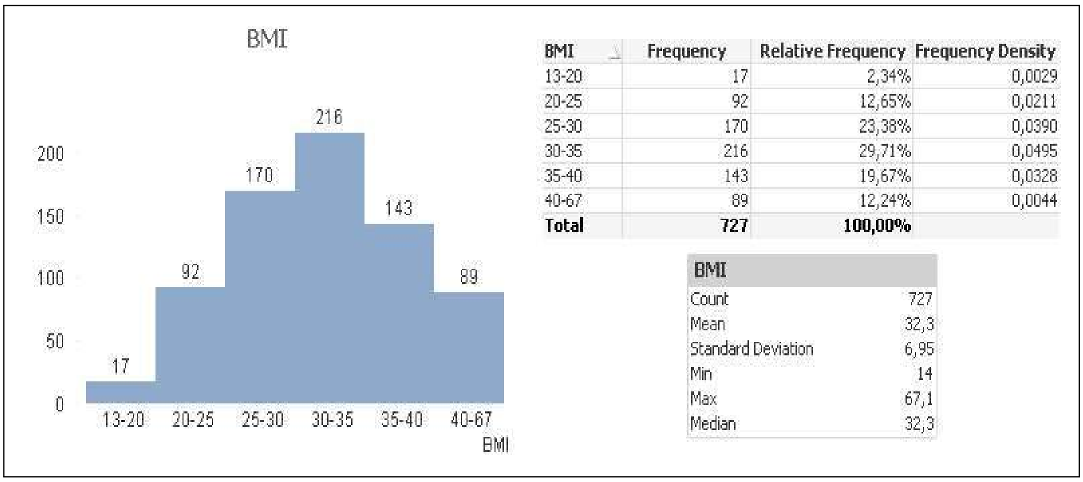
Age	Frequency	Relative Frequency	Frequency Density
21-25	252	34,66%	0,0693
25-35	214	29,44%	0,0294
35-45	144	19,81%	0,0396
45-81	117	16,09%	0,0045
Total	**727**	**100,00%**	

Age	
Count	727
Mean	33,4
Standard Deviation	11,87
Min	21
Max	81
Median	29

Source – Elaborated by the author

The majority of patients, 65%, have age ranging between 21 and 35 years. The mean (33.4) and median (29) are close, keeping into the same range (25-35). The distribution is right-skewed.

Figure 30 – BMI statistical information



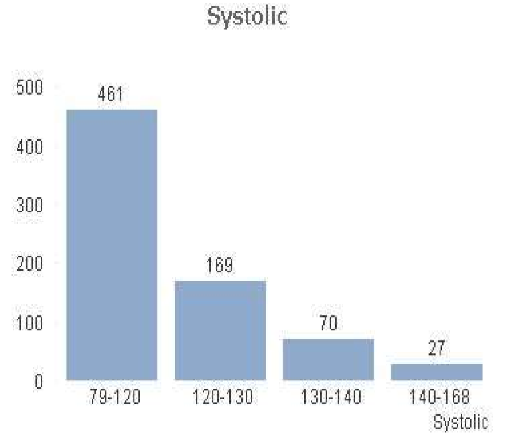
BMI	Frequency	Relative Frequency	Frequency Density
13-20	17	2,34%	0,0029
20-25	92	12,65%	0,0211
25-30	170	23,38%	0,0390
30-35	216	29,71%	0,0495
35-40	143	19,67%	0,0328
40-67	89	12,24%	0,0044
Total	**727**	**100,00%**	

BMI	
Count	727
Mean	32,3
Standard Deviation	6,95
Min	14
Max	67,1
Median	32,3

Source – Elaborated by the author

The majority of patients, 73%, have a body mass index between 25 and 40 kg/m2. The mean and median are the same (32.3), keeping into the range (30-35). The distribution has a standard deviation of 6.95.

Figure 31 – Systolic statistical information



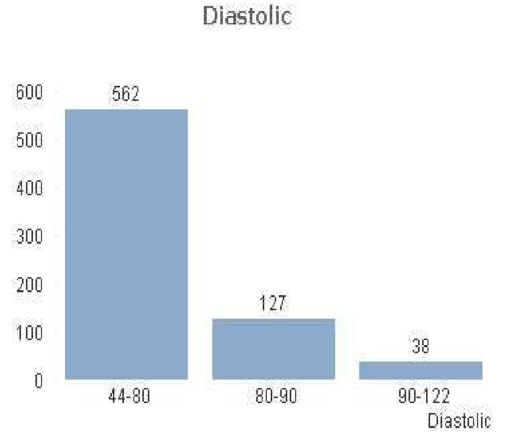
Systolic	Frequency	Relative Frequency	Frequency Density
79-120	461	63,41%	0,0151
120-130	169	23,25%	0,0080
130-140	70	9,63%	0,0088
140-168	27	3,71%	0,0013
Total	**727**	**100,00%**	

Systolic	
Count	727
Mean	115,7
Standard Deviation	13,69
Min	79
Max	168
Median	116

Source – Elaborated by the author

The majority of patients, 63%, have a systolic blood pressure ranging between 79 and 120 mmHg. The mean (115.7) and median (116) are close, keeping into the same range (79-120). The distribution is right-skewed.

Figure 32 – Diastolic statistical information



Diastolic	Frequency	Relative Frequency	Frequency Density
44-80	562	77,30%	0,0209
80-90	127	17,47%	0,0159
90-122	38	5,23%	0,0016
Total	**727**	**100,00%**	

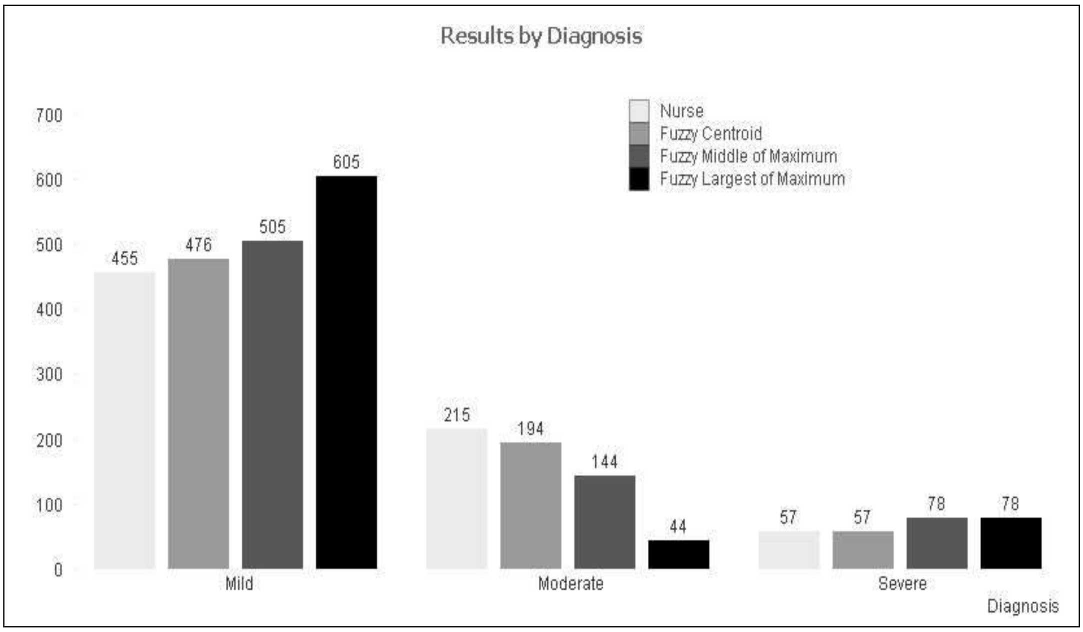
Diastolic	
Count	727
Mean	72,7
Standard Deviation	11,89
Min	44
Max	122
Median	72

Source – Elaborated by the author

The majority of patients, 77%, have a diastolic blood pressure ranging between 44 and 80 mmHg. The mean (72.7) and median (72) are very close, keeping into the same range (44-80). The distribution is right-skewed.

The results achieved by diagnosis, considering nurses work and three defuzzification methods (centroid, middle of maximum and largest of maximum) are shown in Figure 33.

Figure 33 – Achieved results by diagnosis.



Source – Elaborated by the author

Comparing the diagnosis made by nurses with other three methods that used fuzzy logic, it's possible to notice that defuzzification using either Largest of Maximum (LoM) or Middle of Maximum (MoM) methods were too distant of nurse's diagnosis - 455 vs. 605 in Mild, 215 vs. 44 in Moderate, and 57 vs. 78 in Severe for LoM and 455 vs. 505 in Mild, 215 vs. 144 in Moderate, and 57 vs. 78 in Severe for MoM. However, the Centroid method was the closest to the diagnosis made by nurses - 455 vs. 476 in Mild, 215 vs. 194 in Moderate, and 57 vs. 57 in Severe. From this point, we decided to use Centroid as the comparison method with nurses inferences.

Using either fuzzy or nurse evaluation, the severe diagnosis where the same – 57 patients. The moderate diagnosis had a little difference applying those evaluations – 215 for nurses and 194 for fuzzy (difference on 21 results). The mild diagnosis was the most expressive representation, but the difference between the evaluation methods – 476 for fuzzy and 455 for nurses (difference on 21 results) weren't too significant.

Analyzing the universe set of 768 patients on the database, 41 patients data were considered invalid due to some constraints, such as diastolic blood pressure value equals to zero. Table 4 depicts defuzzification methods - Centroid, Middle of Maximum, Largest of Maximum - compared to inferences made by the nurses.

The centroid defuzzification method is the best way to infer blood pressure level, considering this scenario. As shown in figure 34, the remaining universe of 727 patients, we had

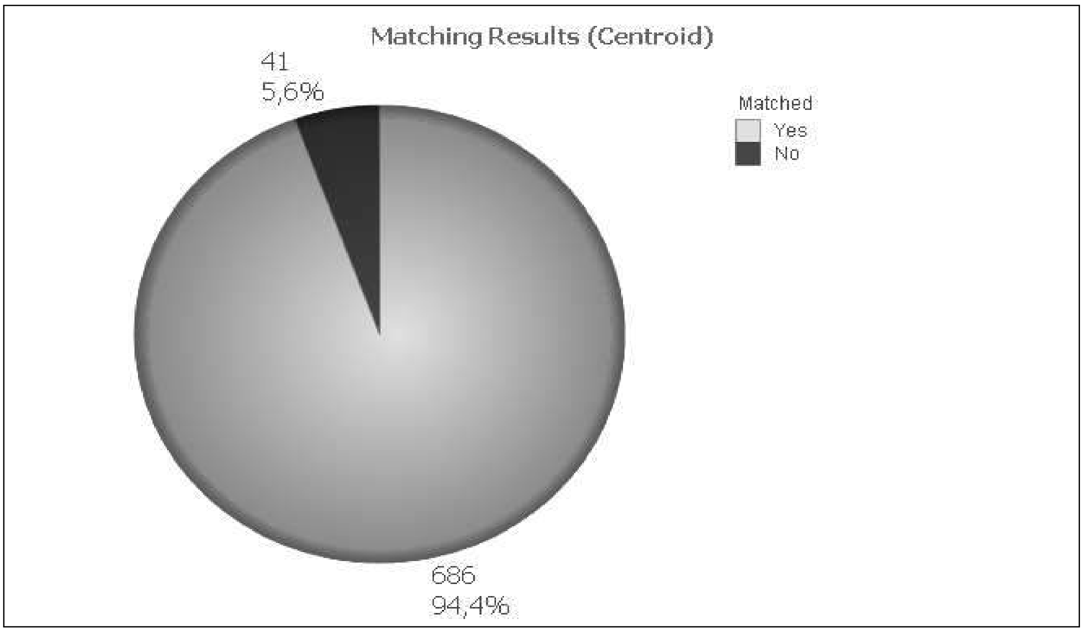
Table 4 – Matching Results Comparision - Inferences using Fuzzy Logic

	Defuzzification - Centroid			Defuzzification - MOM			Defuzzification - LOM		
	Qty	%	Running Time	Qty	%	Running Time	Qty	%	Running Time
Nurse	686	94.40	6.29 sec	640	88.00	6.35 sec	539	74.10	6.85 sec

Source – Elaborated by the author

an amount of 41 where the diagnosis given by the nurses and fuzzy don't match, and 686 that match.

Figure 34 – Matching Results - Centroid Defuzzification.



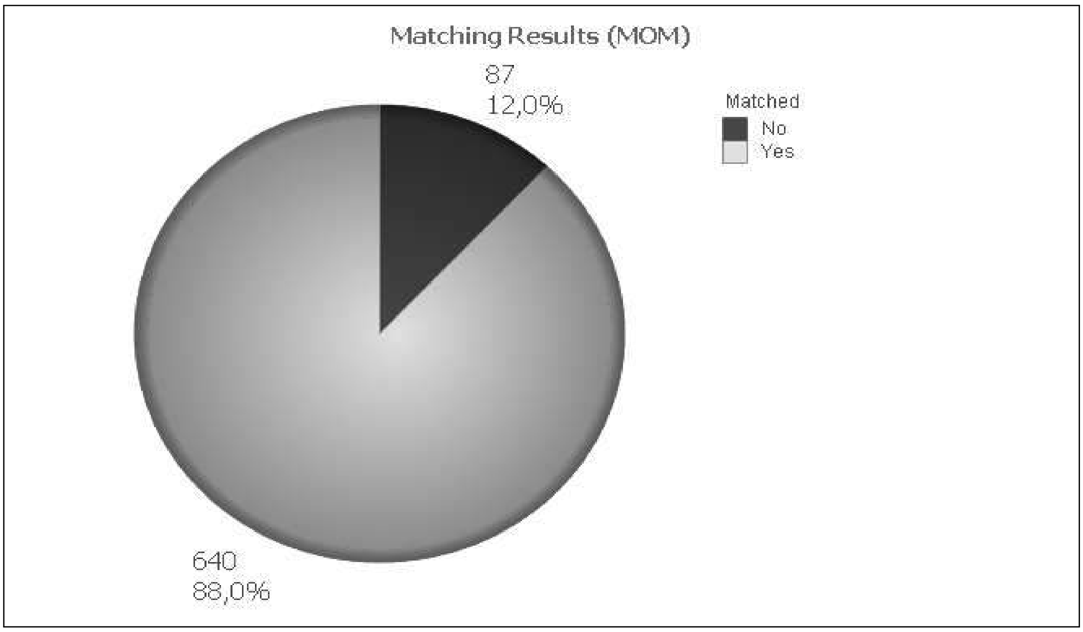
Source – Elaborated by the author

In figure 35, the remaining universe of 727 patients, we had an amount of 87 where the diagnosis given by the nurses and fuzzy don't match, and 640 that match.

In figure 36, the remaining universe of 727 patients, we had an amount of 138 where the diagnosis given by the nurses and fuzzy don't match, and 539 that match.

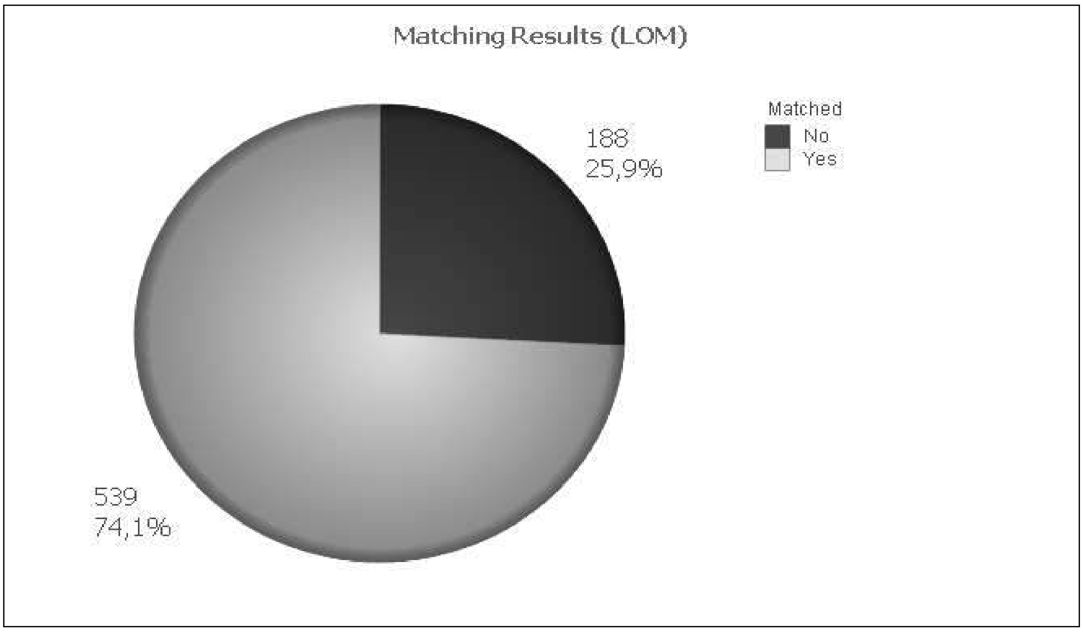
The monitoring agent has taken the results of processing stored in the database and made available through a web page with search filters. From this point, the MAS solution made it possible to cross the data and infer new information and scenarios, when desired by the user.

Figure 35 – Matching Results - Middle of Maximum.



Source – Elaborated by the author

Figure 36 – Matching Results - Largest of Maximum.



Source – Elaborated by the author

6 CONCLUSION AND FUTURE WORKS

Multi-agent systems working together with IoT devices has been increasing in last years. Moreover, applying a certain level of artificial intelligence to some of those agents makes things better, considering the fast and precise diagnosis. It's essential to observe that more and more people are living alone and they should take care of health and quality of life. The industry offers several smart devices focused on health. This work shows an implementation of agents to capture vital signals of user/patient, using affordable hardware devices, and the due treatment of them, inferring the level of blood pressure.

Following this direction, using data obtained from "Pima Indians Diabetes Data Set" on UCI Machine Learning Repository Database (SIGILLITO, 2013) and the application of the implemented model based on fuzzy logic, the diagnosis of blood pressure level was interesting. The results of our solution were validated by two accredited nurses and achieved an accuracy of 94.40% in High Blood Pressure Diagnosis, where 686 results matched.

As future work, we intend to finish the integration and automation of this model that uses fuzzy logic within an agent as proposed in the architecture and improve the code to calculate systolic and diastolic blood pressure from ECG and Photoplethysmography (PPG) sensors. Moreover, we intend to implement others elements of the proposed architecture, such as using a NoSQL database, Machine Learning techniques, and integration with smart bands.

A project was submitted to Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), where we intend to build more comfortable devices to monitor patients with chronic conditions such as Chagas disease, cardiac transplants, etc.

BIBLIOGRAPHY

ACAMPORA G., C. D. J. R. P. V. A. V. A survey on ambient intelligence in health care. **Proceedings of the IEEE**, ieee, v. 101, p. 2470–2494, 2013.

AGING, N. I. on. Aging well in the 21st century: Strategic directions for research on aging. 2016. Disponível em: <http://www.nia.nih.gov/about/aging-well-21st-century-strategic-directions-research-aging/>.

AL-KHATIB, S. M.; STEVENSON, W. G.; ACKERMAN, M. J.; BRYANT, W. J.; CALLANS, D. J.; CURTIS, A. B.; DEAL, B. J.; DICKFELD, T.; FIELD, M. E.; FONAROW, G. C. *et al.* 2017 aha/acc/hrs guideline for management of patients with ventricular arrhythmias and the prevention of sudden cardiac death: a report of the american college of cardiology/american heart association task force on clinical practice guidelines and the heart rhythm society. **Journal of the American College of Cardiology**, Journal of the American College of Cardiology, p. 24390, 2017.

AMBIENT ASSISTED LIVING (AAL). **Success Stories**. 2012. Disponível em: <http://www.aal-europe.eu/success-stories/>.

CARLIER, F.; RENAULT, V. Iot-a, embedded agents for smart internet of things. application on a display wall. In: **2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW)**. [s.n.], 2016. p. 80–83. Disponível em: <https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7811912/>.

CHANDRA, P. S. V. Fuzzy based high blood pressure diagnosis. **International Journal of Advanced Research in Computer Science amp; Technology - IJARCST**, v. 2, 2014.

COMMISSION, E. Research and innovation. **The European Commission Explained**, 2014. Disponível em: <https://europa.eu/european-union/topics_en/>.

CUNO S., G. Y. B. A. H. M. S. A. A. A. . C. R. Management, monitoring and supporting dementia patients at home by aladdin. **Proceedings of the IEEE**, ieee, 2011.

DÍAZ-RODRÍGUEZ, N.; HÄRMÄ, A.; HELAOUI, R.; HUITZIL, I.; BOBILLO, F.; STRACCIA, U. Couch potato or gym addict? semantic lifestyle profiling with wearables and knowledge graphs. In: **Proceedings of the 6th Workshop on Automated Knowledge Base Construction (AKBC 2017), Long Beach (USA)**. [s.n.], 2017. Disponível em: <http://www.akbc.ws/2017/papers/16_paper.pdf>.

FELIZARDO, V.; SOUSA, P.; SABUGUEIRO, D.; ALEXANDRE, C.; COUTO, R.; GARCIA, N.; PIRES, I. E-health: Current status and future trends. p. 302–326, 01 2014.

GROUP, S. O. A. W. **HL7 Version 3 Specification: hData Record Format, Release 1**. 2014. Disponível em: <http://www.hl7.org/documentcenter/private/standards_temp_DA527FF2-1C23-BA17-0C002A2ACD3A2B3B/v3/V3_ITS_HDATA_RF_R1_2014JUN.pdf>.

HAGHI, M.; THUROW, K.; STOLL, R. Wearable devices in medical internet of things: scientific research and commercially available devices. **Healthcare informatics research**, v. 23, n. 1, p. 4–15, 2017. Disponível em: <https://www.ncbi.nlm.nih.gov/pubmed/28261526/>.

HANDFORD, R. **Telefonica reveals future plans following Parkinson's trial**. 2013. Disponível em: <http://www.mobileworldlive.com/telefonica-announces-results-future-plans-followingparkinsons-trial/>.

HARITOU M., G. Y. A. A. X. S. A. A. . B. A. e. a. A technology platform for a novel home care delivery service to patients with dementia. **Journal of Medical Imaging and Health Informatics**, v. 2, p. 1–7, 2012.

IMAI, Y.; AIHARA, A.; OHKUBO, T.; NAGAI, K.; TSUJI, I.; MINAMI, N.; SATOH, H.; HISAMICHI, S. Factors that affect blood pressure variability. a community-based study in ohasama, japan. **American journal of hypertension**, Oxford University Press, v. 10, n. 11, p. 1281–1289, 1997. Disponível em: <https://www.ncbi.nlm.nih.gov/pubmed/9397248/>.

ISLAM, S. M. R.; KWAK, D.; KABIR, M. H.; HOSSAIN, M.; KWAK, K. The internet of things for health care: A comprehensive survey. **IEEE Access**, v. 3, p. 678–708, 2015. ISSN 2169-3536. Disponível em: <https://ieeexplore.ieee.org/document/7113786/>.

KUMAR, S.; AYUB, S. Estimation of blood pressure by using electrocardiogram (ecg) and photo-plethysmogram (ppg). In: IEEE. **Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on**. [S.l.], 2015. p. 521–524.

MALACHIAS, M. 7th brazilian guideline of arterial hypertension: Presentation. **Arquivos Brasileiros de Cardiologia**, scielo, v. 107, p. XV – XIX, 09 2016. ISSN 0066-782X. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0066-782X2016004800001&nrm=iso>.

MQTT. **The MQTT home page**. 2018. Disponível em: <http://mqtt.org/>.

NETO, A. B. L.; ANDRADE, J. P. B.; LOUREIRO, T. C. J.; CAMPOS, G. A. L. d.; FERNANDEZ, M. P. Fuzzy logic applied to ehealth supported by a multi-agent system. **37th NAFIPS Annual Conference**, 07 2018. Disponível em: <https://nafips2018.wordpress.com/>.

NETO, A. B. L.; ANDRADE, J. P. B.; LOUREIRO, T. C. J.; CAMPOS, G. A. L. d.; FERNANDEZ, M. P. A multi-agent system using fuzzy logic applied to ehealth. **9th International Symposium on Ambient Intelligence**, 06 2018. Disponível em: <https://www.isami-conference.net/>.

PANG, Z. **Technologies and Architectures of the Internet-of-Things (IoT) for Health and Well-being**. Tese (Doutorado) — KTH Royal Institute of Technology, 2013. Disponível em: <https://pdfs.semanticscholar.org/222d/206e8fc758c19ac06680db61a555fd6b71ed.pdf>.

PROJECT, R. **Guidance and Awareness Services for Independent Living (Flyer)**. 2012. Disponível em: <http://www.aal-rosetta.eu/Site_rosetta/docs/ROSETTA_Flyer_e_IESE-100118.pdf>.

R-PROJECT. **R-project home page**. 2018. Disponível em: <https://www.r-project.org>.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence A Modern Approach**. [S.l.]: Malaysia; Pearson Education Limited,, 2010. v. 3.

SANTAMARíA, A. F.; RAIMONDO, P.; RANGO, F. D.; SERIANNI, A. A two stages fuzzy logic approach for internet of things (iot) wearable devices. In: **2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)**. [s.n.], 2016. p. 1–6. ISSN 2166-9589. Disponível em: <https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7762553/>.

SHOHAM, Y.; LEYTON-BROWN, K. **Multiagent systems: Algorithmic, game-theoretic, and logical foundations**. [S.l.]: Cambridge University Press, 2010.

SIGILLITO, V. Pima indians diabetes data set. **The Johns Hopkins University Lichman**, UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science, p. 1281–1289, 2013. Disponível em: <http://archive.ics.uci.edu/ml/>.

SPARKFUN. **AD8232 Heart Rate Monitor Hookup Guide**. 2015. Disponível em: <https://learn.sparkfun.com/tutorials/ad8232-heart-rate-monitor-hookup-guide?_ga=2.223652761.320702044.1531164680-1857235492.1531164680/>.

TAMIR, D. E.; RISHE, N. D.; KANDEL, A. **Comprehensive and timely report on fuzzy logic and its applications**. [S.l.]: Springer, 2015. v. 326.

TOADER, C. G. Multi-agent based e-health system. In: **2017 21st International Conference on Control Systems and Computer Science (CSCS)**. [s.n.], 2017. p. 696–700. ISSN 2379-0482. Disponível em: <https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7963998/>.

WADHWA, R.; MEHRA, A.; SINGH, P.; SINGH, M. A pub/sub based architecture to support public healthcare data exchange. In: **2015 7th International Conference on Communication Systems and Networks (COMSNETS)**. [s.n.], 2015. p. 1–6. ISSN 2155-2487. Disponível em: <https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7090755/>.

ZADEH, L. A. Fuzzy sets. **Information and control**, v. 8, p. 3, 1965. Disponível em: <https://www.sciencedirect.com/science/article/pii/S001999586590241X/>.

APPENDIX

APPENDIX A – RULES OF FUZZY INFERENCE SYSTEM

Number	Rule
01	If (sbp is severe) and (dbp is severe) then (hbpr is severe)
02	If (sbp is moderate) and (dbp is moderate) and (age is young) and (bmi is high) then (hbpr is severe)
03	If (sbp is moderate) and (dbp is moderate) and (age is young) and (bmi is vhigh) then (hbpr is severe)
04	If (sbp is mild) and (dbp is severe) and (age is young) and (bmi is high) then (hbpr is severe)
05	If (sbp is mild) and (dbp is severe) and (age is young) and (bmi is vhigh) then (hbpr is severe)
06	If (sbp is moderate) and (dbp is severe) and (age is young) and (bmi is high) then (hbpr is severe)
07	If (sbp is moderate) and (dbp is severe) and (age is young) and (bmi is vhigh) then (hbpr is severe)
08	If (sbp is mild) and (dbp is moderate) and (age is young) and (bmi is high) then (hbpr is moderate)
09	If (sbp is mild) and (dbp is moderate) and (age is young) and (bmi is vhigh) then (hbpr is moderate)
10	If (sbp is mild) and (dbp is moderate) and (age is middl) and (bmi is high) then (hbpr is moderate)
11	If (sbp is mild) and (dbp is moderate) and (age is middl) and (bmi is vhigh) then (hbpr is moderate)
12	If (sbp is moderate) and (dbp is mild) and (age is young) and (bmi is high) then (hbpr is moderate)
13	If (sbp is moderate) and (dbp is mild) and (age is young) and (bmi is vhigh) then (hbpr is moderate)
14	If (sbp is moderate) and (dbp is mild) and (age is middl) and (bmi is high) then (hbpr is moderate)
15	If (sbp is moderate) and (dbp is mild) and (age is middl) and (bmi is vhigh) then (hbpr is moderate)

16	If (sbp is mild) and (dbp is severe) and (age is middl) and (bmi is low) then (hbpr is moderate)
17	If (sbp is mild) and (dbp is severe) and (age is middl) and (bmi is norm) then (hbpr is moderate)
18	If (sbp is mild) and (dbp is severe) and (age is old) and (bmi is low) then (hbpr is moderate)
19	If (sbp is mild) and (dbp is severe) and (age is old) and (bmi is norm) then (hbpr is moderate)
20	If (sbp is mild) and (dbp is severe) and (age is vold) and (bmi is low) then (hbpr is moderate)
21	If (sbp is mild) and (dbp is severe) and (age is vold) and (bmi is norm) then (hbpr is moderate)
22	If (sbp is mild) and (dbp is mild) and (age is young) and (bmi is norm) then (hbpr is mild)
23	If (sbp is mild) and (dbp is mild) and (age is young) and (bmi is high) then (hbpr is mild)
24	If (sbp is mild) and (dbp is mild) and (age is young) and (bmi is vhigh) then (hbpr is mild)
25	If (sbp is mild) and (dbp is mild) and (age is middl) and (bmi is low) then (hbpr is mild)
26	If (sbp is mild) and (dbp is mild) and (age is middl) and (bmi is norm) then (hbpr is mild)
27	If (sbp is mild) and (dbp is mild) and (age is middl) and (bmi is high) then (hbpr is mild)
28	If (sbp is mild) and (dbp is mild) and (age is middl) and (bmi is vhigh) then (hbpr is mild)
29	If (sbp is mild) and (dbp is mild) and (age is old) and (bmi is low) then (hbpr is mild)
30	If (sbp is mild) and (dbp is mild) and (age is old) and (bmi is high) then (hbpr is mild)
31	If (sbp is mild) and (dbp is mild) and (age is old) and (bmi is vhigh) then (hbpr is mild)

32	If (sbp is mild) and (dbp is mild) and (age is vold) and (bmi is low) then (hbpr is mild)
33	If (sbp is mild) and (dbp is mild) and (age is vold) and (bmi is norm) then (hbpr is mild)
34	If (sbp is mild) and (dbp is mild) and (age is vold) and (bmi is high) then (hbpr is mild)
35	If (sbp is mild) and (dbp is moderate) and (age is young) and (bmi is low) then (hbpr is mild)
36	If (sbp is mild) and (dbp is moderate) and (age is young) and (bmi is norm) then (hbpr is mild)
37	If (sbp is mild) and (dbp is moderate) and (age is middl) and (bmi is norm) then (hbpr is mild)
38	If (sbp is mild) and (dbp is moderate) and (age is middl) and (bmi is low) then (hbpr is mild)
39	If (sbp is mild) and (dbp is moderate) and (age is old) and (bmi is high) then (hbpr is mild)
40	If (sbp is mild) and (dbp is moderate) and (age is old) and (bmi is vhigh) then (hbpr is mild)
41	If (sbp is mild) and (dbp is moderate) and (age is vold) and (bmi is low) then (hbpr is mild)
42	If (sbp is mild) and (dbp is moderate) and (age is vold) and (bmi is norm) then (hbpr is mild)
43	If (sbp is mild) and (dbp is moderate) and (age is vold) and (bmi is high) then (hbpr is mild)
44	If (sbp is mild) and (dbp is moderate) and (age is vold) and (bmi is vhigh) then (hbpr is mild)
45	If (sbp is mild) and (dbp is severe) and (age is young) and (bmi is norm) then (hbpr is mild)
46	If (sbp is mild) and (dbp is severe) and (age is middl) and (bmi is high) then (hbpr is mild)
47	If (sbp is mild) and (dbp is severe) and (age is old) and (bmi is high) then (hbpr is mild)

48	If (sbp is mild) and (dbp is severe) and (age is old) and (bmi is vhigh) then (hbpr is mild)
49	If (sbp is mild) and (dbp is severe) and (age is vold) and (bmi is vhigh) then (hbpr is mild)
50	If (sbp is moderate) and (dbp is mild) and (age is young) and (bmi is low) then (hbpr is mild)
51	If (sbp is moderate) and (dbp is mild) and (age is old) and (bmi is low) then (hbpr is mild)
52	If (sbp is moderate) and (dbp is mild) and (age is old) and (bmi is norm) then (hbpr is mild)
53	If (sbp is moderate) and (dbp is mild) and (age is old) and (bmi is high) then (hbpr is mild)
54	If (sbp is moderate) and (dbp is mild) and (age is vold) and (bmi is low) then (hbpr is mild)
55	If (sbp is moderate) and (dbp is mild) and (age is vold) and (bmi is norm) then (hbpr is mild)
56	If (sbp is moderate) and (dbp is mild) and (age is vold) and (bmi is vhigh) then (hbpr is mild)
57	If (sbp is moderate) and (dbp is mild) and (age is middl) and (bmi is low) then (hbpr is mild)
58	If (sbp is moderate) and (dbp is mild) and (age is middl) and (bmi is norm) then (hbpr is mild)
59	If (sbp is moderate) and (dbp is moderate) and (age is young) and (bmi is low) then (hbpr is mild)
60	If (sbp is moderate) and (dbp is moderate) and (age is young) and (bmi is norm) then (hbpr is mild)
61	If (sbp is moderate) and (dbp is moderate) and (age is middl) and (bmi is low) then (hbpr is mild)
62	If (sbp is moderate) and (dbp is moderate) and (age is middl) and (bmi is norm) then (hbpr is mild)
63	If (sbp is moderate) and (dbp is moderate) and (age is middl) and (bmi is high) then (hbpr is mild)

64	If (sbp is moderate) and (dbp is moderate) and (age is middl) and (bmi is vhigh) then (hbpr is mild)
65	If (sbp is moderate) and (dbp is moderate) and (age is old) and (bmi is low) then (hbpr is mild)
66	If (sbp is moderate) and (dbp is moderate) and (age is old) and (bmi is norm) then (hbpr is mild)
67	If (sbp is moderate) and (dbp is moderate) and (age is old) and (bmi is high) then (hbpr is mild)
68	If (sbp is moderate) and (dbp is moderate) and (age is old) and (bmi is vhigh) then (hbpr is mild)
69	If (sbp is moderate) and (dbp is moderate) and (age is vold) and (bmi is low) then (hbpr is mild)
70	If (sbp is moderate) and (dbp is moderate) and (age is vold) and (bmi is norm) then (hbpr is mild)
71	If (sbp is moderate) and (dbp is moderate) and (age is vold) and (bmi is high) then (hbpr is mild)
72	If (sbp is moderate) and (dbp is moderate) and (age is vold) and (bmi is vhigh) then (hbpr is mild)
73	If (sbp is moderate) and (dbp is severe) and (age is young) and (bmi is low) then (hbpr is mild)
74	If (sbp is moderate) and (dbp is severe) and (age is young) and (bmi is norm) then (hbpr is mild)
75	If (sbp is moderate) and (dbp is severe) and (age is middl) and (bmi is low) then (hbpr is mild)
76	If (sbp is moderate) and (dbp is severe) and (age is middl) and (bmi is norm) then (hbpr is mild)
77	If (sbp is moderate) and (dbp is severe) and (age is middl) and (bmi is high) then (hbpr is mild)
78	If (sbp is moderate) and (dbp is severe) and (age is middl) and (bmi is vhigh) then (hbpr is mild)
79	If (sbp is moderate) and (dbp is severe) and (age is old) and (bmi is low) then (hbpr is mild)

80	If (sbp is moderate) and (dbp is severe) and (age is old) and (bmi is norm) then (hbpr is mild)
81	If (sbp is moderate) and (dbp is severe) and (age is old) and (bmi is high) then (hbpr is mild)
82	If (sbp is moderate) and (dbp is severe) and (age is old) and (bmi is vhigh) then (hbpr is mild)
83	If (sbp is moderate) and (dbp is severe) and (age is vold) and (bmi is high) then (hbpr is mild)
84	If (sbp is moderate) and (dbp is severe) and (age is vold) and (bmi is vhigh) then (hbpr is mild)
85	If (sbp is severe) and (dbp is mild) and (age is young) and (bmi is high) then (hbpr is mild)
86	If (sbp is severe) and (dbp is mild) and (age is young) and (bmi is vhigh) then (hbpr is mild)
87	If (sbp is severe) and (dbp is mild) and (age is middl) and (bmi is low) then (hbpr is mild)
88	If (sbp is severe) and (dbp is mild) and (age is middl) and (bmi is norm) then (hbpr is mild)
89	If (sbp is severe) and (dbp is mild) and (age is middl) and (bmi is high) then (hbpr is mild)
90	If (sbp is severe) and (dbp is mild) and (age is middl) and (bmi is vhigh) then (hbpr is mild)
91	If (sbp is severe) and (dbp is mild) and (age is old) and (bmi is norm) then (hbpr is mild)
92	If (sbp is severe) and (dbp is mild) and (age is vold) and (bmi is low) then (hbpr is mild)
93	If (sbp is severe) and (dbp is mild) and (age is vold) and (bmi is norm) then (hbpr is mild)
94	If (sbp is severe) and (dbp is mild) and (age is vold) and (bmi is high) then (hbpr is mild)
95	If (sbp is severe) and (dbp is moderate) and (age is young) and (bmi is low) then (hbpr is mild)

96	If (sbp is severe) and (dbp is moderate) and (age is young) and (bmi is norm) then (hbpr is mild)
97	If (sbp is severe) and (dbp is moderate) and (age is young) and (bmi is vhigh) then (hbpr is mild)
98	If (sbp is severe) and (dbp is moderate) and (age is young) and (bmi is high) then (hbpr is mild)
99	If (sbp is severe) and (dbp is moderate) and (age is middl) and (bmi is high) then (hbpr is mild)
100	If (sbp is severe) and (dbp is moderate) and (age is middl) and (bmi is vhigh) then (hbpr is mild)
101	If (sbp is severe) and (dbp is moderate) and (age is old) and (bmi is low) then (hbpr is mild)
102	If (sbp is severe) and (dbp is moderate) and (age is old) and (bmi is norm) then (hbpr is mild)
103	If (sbp is severe) and (dbp is moderate) and (age is old) and (bmi is high) then (hbpr is mild)
104	If (sbp is severe) and (dbp is moderate) and (age is old) and (bmi is vhigh) then (hbpr is mild)
105	If (sbp is severe) and (dbp is moderate) and (age is vold) and (bmi is low) then (hbpr is mild)
106	If (sbp is severe) and (dbp is moderate) and (age is vold) and (bmi is norm) then (hbpr is mild)
107	If (sbp is severe) and (dbp is moderate) and (age is vold) and (bmi is high) then (hbpr is mild)
108	If (sbp is severe) and (dbp is moderate) and (age is vold) and (bmi is vhigh) then (hbpr is mild)

APPENDIX B – MOBILE AGENT - CATCHING ECG SIGNAL

```
# MOBILE AGENT
# Sensor: Read ECG signal of the patient
# Process: Check the thresholds of the signal
# Actuator: Send the information through nodeMCU

# Connecting to Wi-Fi using nodeMCU
WiFiClient espClient;                                           // Instance of WiFiClient
PubSubClient client(espClient);

# Connecting to MQTT
//Coloque os valores padroes aqui, porem na interface eles poderao ser substituidos.
#define servidor_mqtt              ""  //MQTT server URL
#define servidor_mqtt_porta        ""  //Server Port
#define servidor_mqtt_usuario      ""  //User
#define servidor_mqtt_senha        ""  //Password
#define mqtt_topico_sub            "esp8266/pincmd"    // Topic

void setup() {
  // initialize the serial communication:
  Serial.begin(9600);
  pinMode(10, INPUT); // Setup for leads off detection LO +
  pinMode(11, INPUT); // Setup for leads off detection LO -
}

void loop() {

  if((digitalRead(10) == 1)||(digitalRead(11) == 1)){
    Serial.println("!");
```

```
  }
  else{
    // send the value of analog input 0:
      Serial.println(analogRead(A0));
  }
  //Wait for a bit to keep serial data from saturating
  delay(1);
}
```

APPENDIX C – PROCESSING AGENT - GET DATA FROM MQTT AND STORE INTO DATABASE

Source Code 4 – GET DATA FROM MQTT AND STORE INTO DATABASE.

```python
# Sensor: Get patients  data from MQTT according to patient
    ID and type of vital signal
# Process: Prepare SQL command to insert into database
# Actuator: Connect and insert data into database

import MySQLdb

# Open database connection
db = MySQLdb.connect("localhost","root","sssddfff","ehealth
   " )
# prepare a cursor object using cursor() method
cursor = db.cursor()

# Publish on MQTT
import paho.mqtt.client as paho
import time
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Connected to broker")
        global Connected                #Use global
           variable
        Connected = True                #Signal connection
    else:
        print("Connection failed")

def on_message(client, userdata, message):
    receivedmsg = str(message.payload)
    print("Message received: "  + receivedmsg)
```

```
    msgarray = receivedmsg.split(";")
    patient=msgarray[0]
    patient=patient[patient.index("=")+1:]
    sistolic=msgarray[1]
    sistolic=sistolic[sistolic.index("=")+1:]
    sistolic_aux=int(sistolic)*10
    sistolic=str(sistolic_aux)
    diastolic=msgarray[2]
    diastolic=diastolic[diastolic.index("=")+1:]
    bmi=msgarray[3]
    bmi=bmi[bmi.index("=")+1:]
    age=msgarray[4]
    age=age[age.index("=")+1:]
    dateinsert = msgarray[5]
    dateinsert=dateinsert[:-1]
    dateinsert=dateinsert[dateinsert.index("=")+1:]
    print(patient)
    print(sistolic)
    print(diastolic)
    print(bmi)
    print(age)
    print(dateinsert)
    comando = " INSERT INTO input2process (V1, V2, V3, V4, V5, V6, V7) " + \
                  " VALUES " + \
                "( "+patient+" , "+sistolic+" , "+diastolic+" ,"+bmi+" , "+ age + \
                " , WaitingProc , "+dateinsert+" )"
    print(comando)
    cursor.execute(comando)
    db.commit()
    #cursor.close()
```

```
Connected = False
client = paho.Client("Python")
client.username_pw_set("lpjyifjc", "iBt39mCeuIDH")
client.on_connect= on_connect                          #attach
   function to callback
client.on_message= on_message
client.connect("m14.cloudmqtt.com", 15024)
client.loop_start()        #start the loop

while Connected != True:    #Wait for connection
    time.sleep(0.1)

client.subscribe("Medicao")

try:
    while True:
        time.sleep(1)

except KeyboardInterrupt:
    print("exiting")
    client.disconnect()
    client.loop_stop()
    # disconnect from server
    db.close()
```

APPENDIX D – PROCESSING AGENT - FUZZY INFERENCE SYSTEM

Source Code 5 – FUZZY INFERENCE SYSTEM.

```
# PROCESSING AGENT - FUZZY.R
# Sensor: Get patients  data from database where status = Waiting Process
# Process: Apply Fuzzy Inference System
# Actuator: Insert processing result into database

library(FuzzyR)
#a = newfis( hyper , fisType = "mamdani", defuzzMethod = "mom")   # Middle of Maximum
#arquivo_saida = "databp_mamdani_middle_of_maximum.csv"
#a = newfis( hyper , fisType = "mamdani", defuzzMethod = "lom") # Largest of Maximum
#arquivo_saida = "databp_mamdani_largest_of_maximum.csv"
a = newfis( hyper , fisType = "mamdani", defuzzMethod = "centroid")
arquivo_saida = "databp_mamdani_centroid_2.csv"
a = newfis( hyper )

################### Variables / Types ####################################

#input
a = addvar(a,  input ,  sbp , c(0, 2500));
a = addvar(a,  input ,  dbp , c(0, 130));
a = addvar(a,  input ,  age , c(0, 100));
a = addvar(a,  input ,  bmi , c(0, 48));

#output
a = addvar(a,  output ,  hbpr , c(0, 30));
```

```

######################### Membership Functions ##############################

a=addmf(a, input ,1, mild , trapmf ,c(-1, 1, 1100, 1200))
a=addmf(a, input ,1, mode , trapmf ,c(1100, 1200, 1300, 1400))
a=addmf(a, input ,1, seve , trapmf ,c(1300, 1400, 1800, 2000))
a=addmf(a, input ,2, mild , trapmf ,c(-16, 0, 75, 80))
a=addmf(a, input ,2, mode , trapmf ,c(70, 80, 85, 90))
a=addmf(a, input ,2, seve , trapmf ,c(80, 90, 130, 140))
a=addmf(a, input ,3, young , trapmf ,c(-1, 10, 20, 30));
a=addmf(a, input ,3, middl , trapmf ,c(25, 30, 40, 45));
a=addmf(a, input ,3, old , trapmf ,c(40, 45, 55, 60));
a=addmf(a, input ,3, vold , trapmf ,c(55, 60, 100, 120));
a=addmf(a, input ,4, low , trapmf ,c(0, 12, 14, 20));
a=addmf(a, input ,4, norm , trapmf ,c(15, 17, 25, 28));
a=addmf(a, input ,4, high , trapmf ,c(26, 29, 36, 40));
a=addmf(a, input ,4, vhigh , trapmf ,c(36, 40, 48, 50));
a=addmf(a, output ,1, mild , trapmf ,c(-3, 0, 7.5, 11.25));
a=addmf(a, output ,1, mode , trapmf ,c(7.5, 11.25, 18.75, 22.5));
a=addmf(a, output ,1, seve , trapmf ,c(18.75, 22.5, 30, 33));

############################### Rule Base ###########################

ruleList = rbind(

  #If SBP is severe AND DBP is severe THEN HBPR is severe
  c(3, 3, 0, 0, 3, 1, 1),
```

```
#If SBP is moderate AND DBP is moderate AND Age is young
   AND BMI is (HIGH OR Very high)THEN HBPR is severe.
c(2, 2, 1, 3, 3, 1, 1),
c(2, 2, 1, 4, 3, 1, 1),
#If SBP is (mild OR moderate) AND DBP is severe  AND Age
   is young  AND BMI is (HIGH OR Very high )THEN HBPR is
   severe.
c(1, 3, 1, 3, 3, 1, 1),
c(1, 3, 1, 4, 3, 1, 1),
c(2, 3, 1, 3, 3, 1, 1),
c(2, 3, 1, 4, 3, 1, 1),
#If SBP is mild AND DBP is moderate AND Age is (young OR
   middle-aged) AND BMI is (HIGH OR Very high )THEN HBPR
   is moderate.
c(1, 2, 1, 3, 2, 1, 1),
c(1, 2, 1, 4, 2, 1, 1),
c(1, 2, 2, 3, 2, 1, 1),
c(1, 2, 2, 4, 2, 1, 1),
#If SBP is moderate AND DBP is mild AND Age is (young OR
   middle-aged) AND BMI is (HIGH OR Very high )THEN HBPR
   is moderate.
c(2, 1, 1, 3, 2, 1, 1),
c(2, 1, 1, 4, 2, 1, 1),
c(2, 1, 2, 3, 2, 1, 1),
c(2, 1, 2, 4, 2, 1, 1),
#If SBP is mild AND DBP is severe AND Age is ( middle-
   aged OR old OR very old) AND BMI is (low OR normal)
   THEN HBPR is moderate.
c(1, 3, 2, 1, 2, 1, 1),
c(1, 3, 2, 2, 2, 1, 1),
c(1, 3, 3, 1, 2, 1, 1),
c(1, 3, 3, 2, 2, 1, 1),
c(1, 3, 4, 1, 2, 1, 1),
```

```
c(1, 3, 4, 2, 2, 1, 1),
#mild
c(1, 1, 1, 2, 1, 1, 1),
c(1, 1, 1, 3, 1, 1, 1),
c(1, 1, 1, 4, 1, 1, 1),
c(1, 1, 2, 1, 1, 1, 1),
c(1, 1, 2, 2, 1, 1, 1),
c(1, 1, 2, 3, 1, 1, 1),
c(1, 1, 2, 4, 1, 1, 1),
c(1, 1, 3, 1, 1, 1, 1),
c(1, 1, 3, 3, 1, 1, 1),
c(1, 1, 3, 4, 1, 1, 1),
c(1, 1, 4, 1, 1, 1, 1),
c(1, 1, 4, 2, 1, 1, 1),
c(1, 1, 4, 3, 1, 1, 1),
c(1, 2, 1, 1, 1, 1, 1),
c(1, 2, 1, 2, 1, 1, 1),
c(1, 2, 2, 2, 1, 1, 1),
c(1, 2, 2, 1, 1, 1, 1),
c(1, 2, 3, 3, 1, 1, 1),
c(1, 2, 3, 4, 1, 1, 1),
c(1, 2, 4, 1, 1, 1, 1),
c(1, 2, 4, 2, 1, 1, 1),
c(1, 2, 4, 3, 1, 1, 1),
c(1, 2, 4, 4, 1, 1, 1),
c(1, 3, 1, 2, 1, 1, 1),
c(1, 3, 2, 3, 1, 1, 1),
c(1, 3, 3, 3, 1, 1, 1),
c(1, 3, 3, 4, 1, 1, 1),
c(1, 3, 4, 4, 1, 1, 1),
c(2, 1, 1, 1, 1, 1, 1),
c(2, 1, 3, 1, 1, 1, 1),
c(2, 1, 3, 2, 1, 1, 1),
```

```
c(2, 1, 3, 3, 1, 1, 1),
c(2, 1, 4, 1, 1, 1, 1),
c(2, 1, 4, 2, 1, 1, 1),
c(2, 1, 4, 4, 1, 1, 1),
c(2, 1, 2, 1, 1, 1, 1),
c(2, 1, 2, 2, 1, 1, 1),
c(2, 2, 1, 1, 1, 1, 1),
c(2, 2, 1, 2, 1, 1, 1),
c(2, 2, 2, 1, 1, 1, 1),
c(2, 2, 2, 2, 1, 1, 1),
c(2, 2, 2, 3, 1, 1, 1),
c(2, 2, 2, 4, 1, 1, 1),
c(2, 2, 3, 1, 1, 1, 1),
c(2, 2, 3, 2, 1, 1, 1),
c(2, 2, 3, 3, 1, 1, 1),
c(2, 2, 3, 4, 1, 1, 1),
c(2, 2, 4, 1, 1, 1, 1),
c(2, 2, 4, 2, 1, 1, 1),
c(2, 2, 4, 3, 1, 1, 1),
c(2, 2, 4, 4, 1, 1, 1),
c(2, 3, 1, 1, 1, 1, 1),
c(2, 3, 1, 2, 1, 1, 1),
c(2, 3, 2, 1, 1, 1, 1),
c(2, 3, 2, 2, 1, 1, 1),
c(2, 3, 2, 3, 1, 1, 1),
c(2, 3, 2, 4, 1, 1, 1),
c(2, 3, 3, 1, 1, 1, 1),
c(2, 3, 3, 2, 1, 1, 1),
c(2, 3, 3, 3, 1, 1, 1),
c(2, 3, 3, 4, 1, 1, 1),
c(2, 3, 4, 3, 1, 1, 1),
c(2, 3, 4, 4, 1, 1, 1),
c(3, 1, 1, 3, 1, 1, 1),
```

```
  c(3, 1, 1, 4, 1, 1, 1),
  c(3, 1, 2, 1, 1, 1, 1),
  c(3, 1, 2, 2, 1, 1, 1),
  c(3, 1, 2, 3, 1, 1, 1),
  c(3, 1, 2, 4, 1, 1, 1),
  c(3, 1, 3, 2, 1, 1, 1),
  c(3, 1, 4, 1, 1, 1, 1),
  c(3, 1, 4, 2, 1, 1, 1),
  c(3, 1, 4, 3, 1, 1, 1),
  c(3, 2, 1, 1, 1, 1, 1),
  c(3, 2, 1, 2, 1, 1, 1),
  c(3, 2, 1, 4, 1, 1, 1),
  c(3, 2, 1, 3, 1, 1, 1),
  c(3, 2, 2, 3, 1, 1, 1),
  c(3, 2, 2, 4, 1, 1, 1),
  c(3, 2, 3, 1, 1, 1, 1),
  c(3, 2, 3, 2, 1, 1, 1),
  c(3, 2, 3, 3, 1, 1, 1),
  c(3, 2, 3, 4, 1, 1, 1),
  c(3, 2, 4, 1, 1, 1, 1),
  c(3, 2, 4, 2, 1, 1, 1),
  c(3, 2, 4, 3, 1, 1, 1),
  c(3, 2, 4, 4, 1, 1, 1)
)

a <- addrule(a, ruleList)

#plotmf(a, "input", 1)
#plotmf(a, "output", 1)

#showGUI(a)

#bp = as.matrix(read.csv("bp-data.csv"))
```

```
#bp = bp[,-1]

# Connecting do eHealth Mysql database
library(DBI)
library(RMySQL)
mydrv <- dbDriver("MySQL")
con <- dbConnect(mydrv, dbname= ehealth , host= localhost , port=3306, user= root , password= af250974 )

#reading data from table with assinged status  WaitingProc -> Waiting Process
res <- dbSendQuery(con, "SELECT V2, V3, V4, V5 FROM input2process WHERE V6 =  WaitingProc ")
bp = as.matrix(dbFetch(res, n=-1))
#bp = bp[,-1]
dbClearResult(res)

list_output = c()

for(i in 1:nrow(bp)){
  input = bp[i,]
  age = input[4]
  input[4] = input[3]
  input[3] = age
  list_output[i] = evalfis(input, a)

}

tb = list_output
for(i in 1:length(tb)){

  value = as.numeric(tb[i])
  if(value<10){
```

```
    c = "low"
  }
  if(value>20){
    c = "high"
  }
  if(value<20 && value>10){
    c = "normal"
  }
    tb[i] = c
    c = 0
}

final = matrix(0, nrow = length(tb), ncol = 8)

final[1:length(tb), 2:5] = bp

final[,1] = c(1:length(tb))
final[,6] = tb
final[,7] = rep("Processed", length(tb))
final[,8] = rep(toString(Sys.time()), length(tb))

write.csv(final, "databp_mamdani.csv")

# save on mysql
df_test <- as.data.frame(final)
dbWriteTable(con, "results2", df_test,
  field.types = NULL, row.names = FALSE, overwrite = FALSE,
  append = TRUE, allow.keywords = FALSE)
```

APPENDIX E – PROCESSING AGENT - PUBLISH ALERTS ON MQTT

Source Code 6 – PUBLISH ALERTS ON MQTT.

```
# PROCESSING AGENT - write2MQTT.py
# Sensor: Read patients  data with  Severe  blood pressure level
# Process: Prepare message for each patient
# Actuator: Publish information on  ALERT  MQTT topic

#Read data from MySQL
import MySQLdb

# Open database connection
db = MySQLdb.connect("localhost","root","ssdddsa","ehealth")

# prepare a cursor object using cursor() method
cursor = db.cursor()

# execute SQL query using execute() method.
cursor.execute("SELECT * from results2 where V6 =  Severe ")

# Preparing in order to Publish on MQTT
import paho.mqtt.client as paho
import time
client = paho.Client()
#client.on_publish = on_publish
client.username_pw_set("lpjyifjc", "iBt39mCeuIDH")
client.connect("m14.cloudmqtt.com", 15024)

# Reading rows and publishing on  ALERT  MQTT topic
```

```
for row in cursor:
    result_print = "Patient: "+str(row[1])+ "; Result: "+
       str(row[5])
    client.publish("ALERT_"+str(row[1]), result_print, qos
       =1)
    time.sleep(0.05)
cursor.close()
# disconnect from server
db.close()
```