

# Practical Applications of Distributed IoT Systems

Iván García-Magariño

Department of Software Engineering and Artificial Intelligence

Complutense University of Madrid, Spain

[igarciam@ucm.es](mailto:igarciam@ucm.es)

# Outline

- Introduction
- Distributed systems for measuring IoT devices for measuring memory in Activity of Daily Life
- Two Case Studies:
  - Assessment of Navigation patterns for measuring memory and disorientation in 3D simulations
  - Real and novel smart cupboard in kitchen for measuring Memory
- Application of Human-centric Artificial Intelligence

# Introduction

## **Problem:**

- Alzheimer is one of the most prevalent disorders
- The main symptom is loss of memory
- Measuring memory requires effort from the subjects
- Continuous Tracking memory usually requires a high level of commitment

## **Proposed solution:**

- Internet of Things (IoT) devices for tracking memory of people by just analyzing their daily activities

# IoT for Measuring Activities in Daily Life

## **Actions that they reflect they forget something:**

- where they left some keys, their shoes, clothes or food
- whether they have changed their clothes
- whether they have had a shower

## **Most of these actions can be tracked by IoT devices in houses**

- Presence Sensors
- Smart cupboards
- Smart wardrobes
- Smart taps

# Algorithms for assessing memory and detecting disorientation

## Reference:

García-Magariño, I., Cárdenas, M., Gómez-Sanz, J., & Pérez Díez (2019). Framework-supported mechanism of testing algorithms for assessing memory and detecting disorientation from IoT sensors . In 5th IEEE World Forum on Internet of Things (WF-IoT) 2019. Limerick, Ireland, April 15-18 2019. IEEE

## Main aspects:

- Use of Presence sensors
- Detecting Navigation Patterns

# Navigation patterns simulated in AIDE 3D simulation platform



**AIDE (developed by  
Grasia research  
group)**

---

**Algorithm 1** Algorithm for measuring memory with presence sensors with pattern  $P_1$ 

---

```
1: procedure INITIALIZEMEMORYTRACKINGSYSTEM( )
2:   sum  $\leftarrow$  0
3:   queue  $\leftarrow$  new CircularQueue()
4: procedure HANDLESENSORINFORMATION( $s_i, \Delta t_i, \Delta t_{s,i}$ )
5:   queue.add ( $\Delta t_i$ )
6:   sum  $\leftarrow$  sum +  $\Delta t_i$ 
7:   if queue.length >  $n_{p,1}$  then
8:     oldT  $\leftarrow$  queue.begin
9:     queue.removeBegin()
10:    sum  $\leftarrow$  sum - oldT
11:    if (sum <  $t_{p,1}$ ) then
12:      notifyMemoryPattern ( $p_1$ )
```

---

---

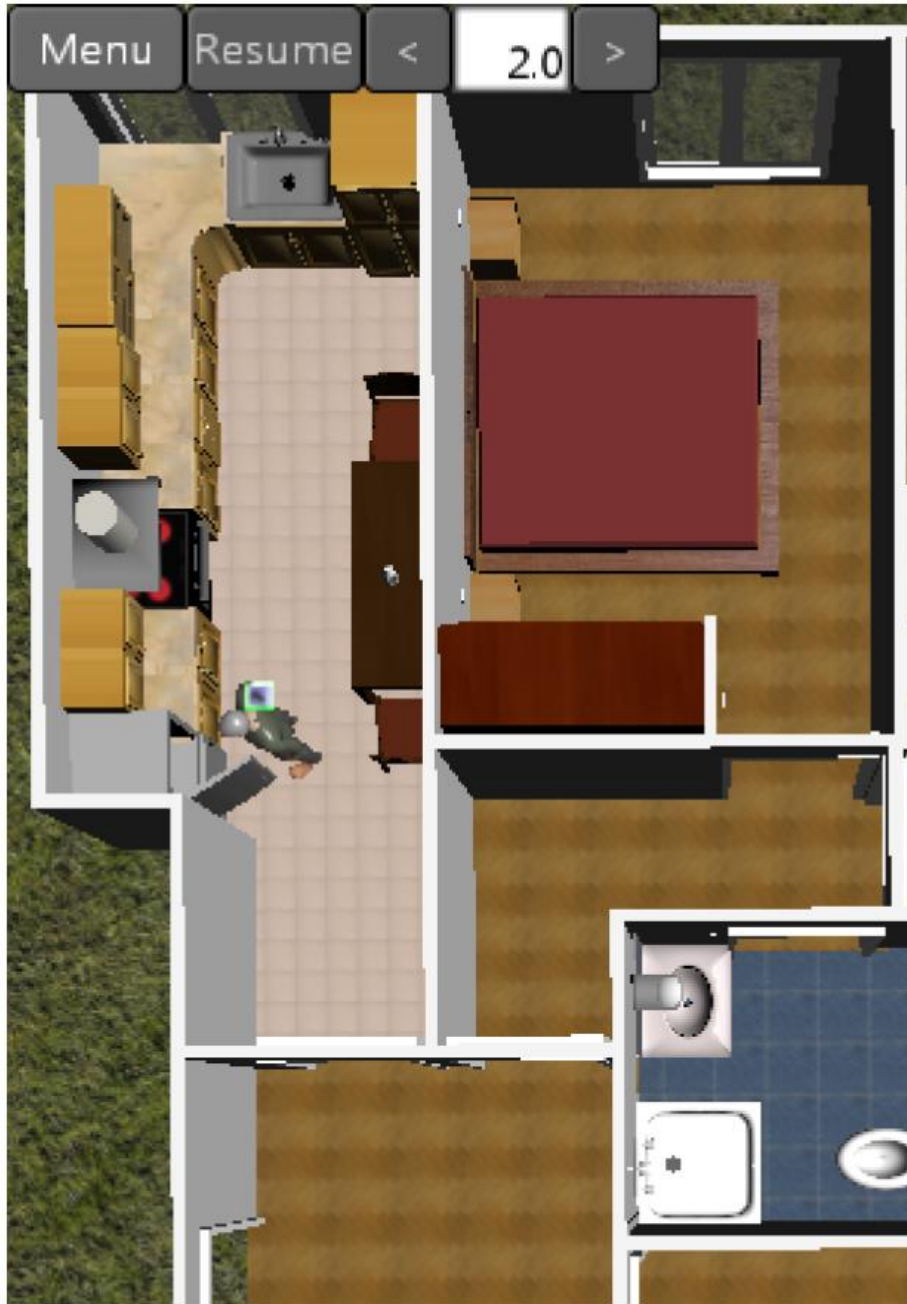
**Algorithm 2** Algorithm for measuring memory with presence sensors with pattern  $P_1$ 

---

```
1: procedure INITIALIZEMEMORYTRACKINGSYSTEM( )
2:   returningTimes gets new List[|S|]
3:   for  $i \in [0, |S|-1]$  do
4:     returningTimes[i]  $\leftarrow$  new List();
5: procedure HANDLESENSORINFORMATION( $s_i, \Delta t_i, \Delta t_{s,i}$ )
6:   returningTimes[ $s_i$ ].add ( $\Delta t_{s,i}$ )
7:   if (returningTimes[ $s_i$ ].length >  $n_{p,2}$ ) and
8: (returningTimes[ $s_i$ ].median <  $t_{p,2}$ ) then
9:     notifyMemoryPattern ( $p_2$ )
```

---





Example of  
the avatar  
opening the  
fridge

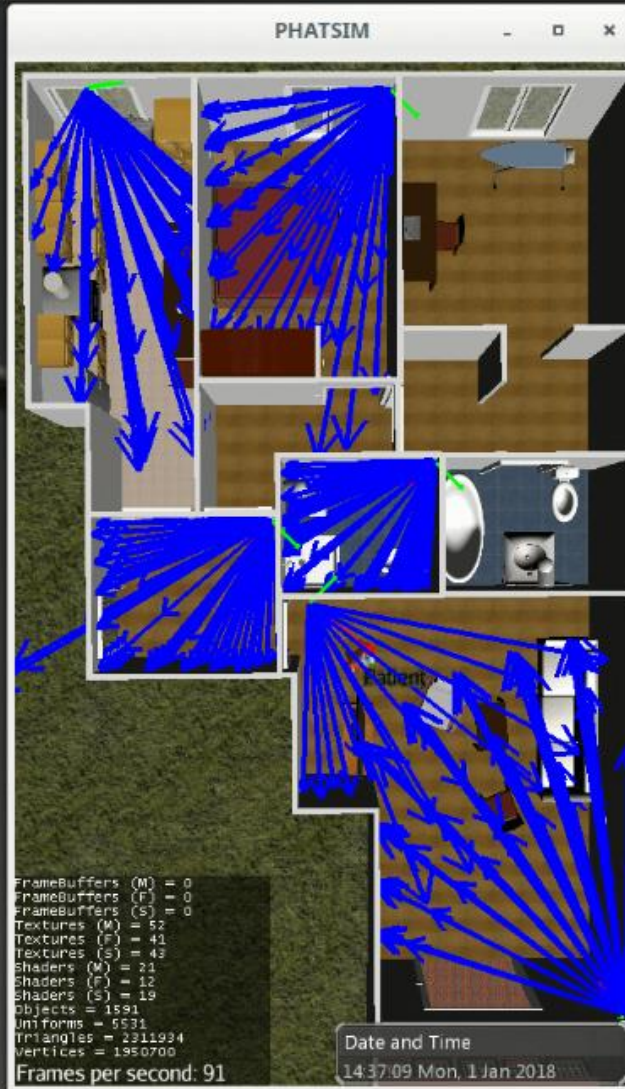
## Sensor Monitoring

Presence Sensor

Id: PreSen-Bedroom1-1 Presence: NO TS: 14:30:21 1/1/2018

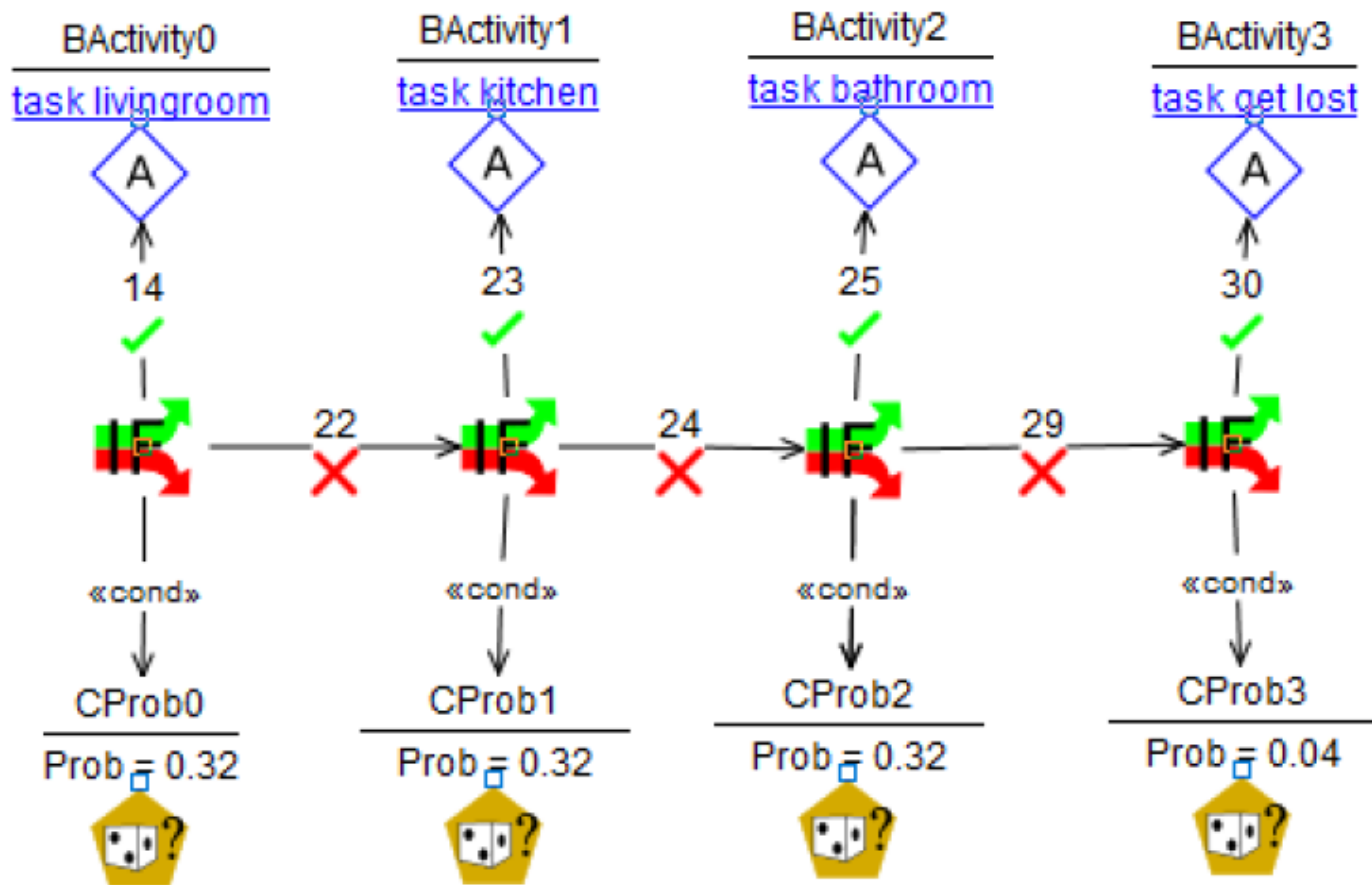
Presence Sensor

Id: PreSen-Kitchen-1 Presence: NO TS: 14:36:47 1/1/2018

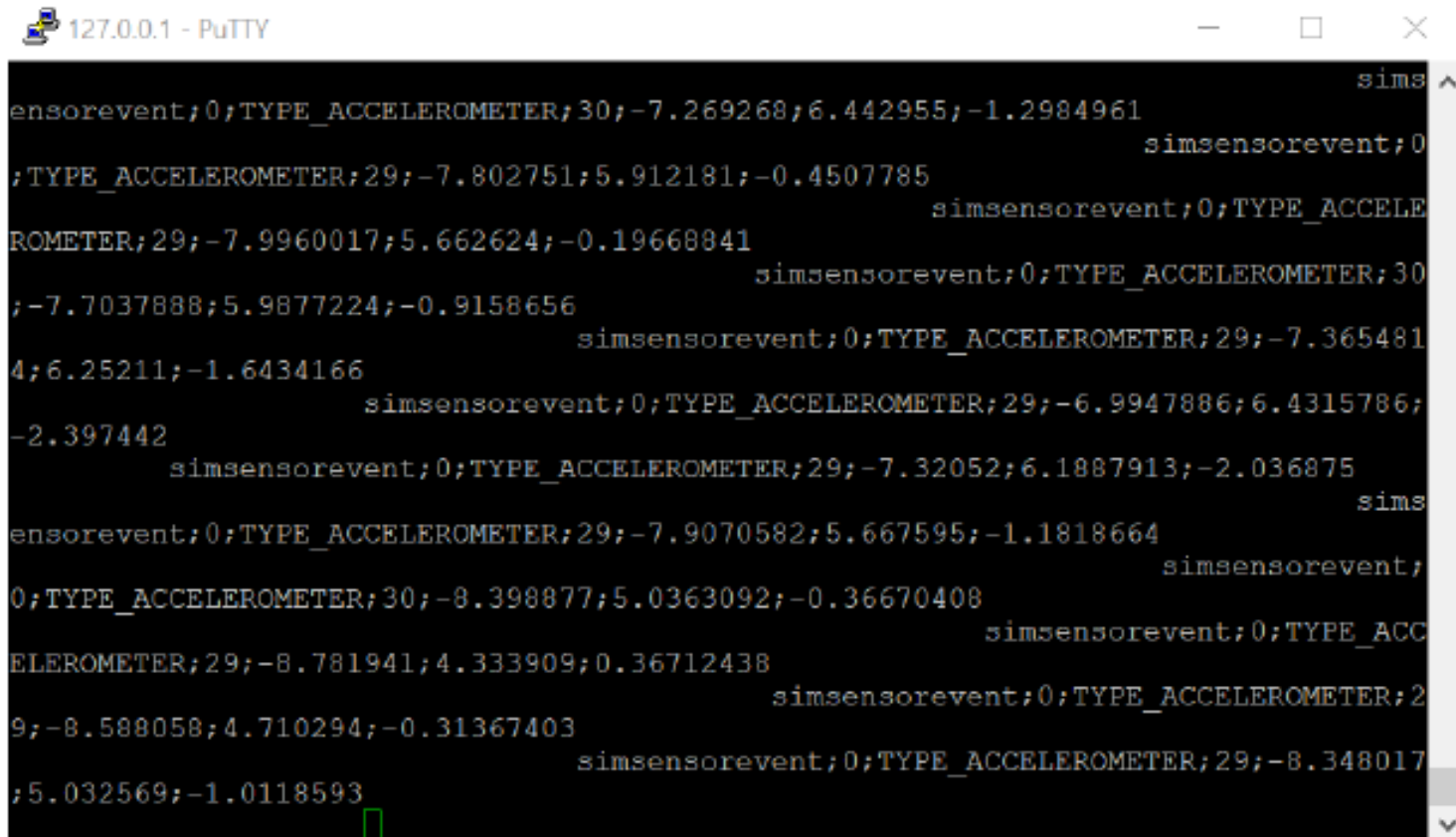


# Presence Sensors in simulations

# Modelling avatar behaviors



# Example of information transmitted to ports in simulations



The image shows a PuTTY terminal window titled "127.0.0.1 - PuTTY". The terminal displays a series of simulation data outputs. Each output line consists of a sensor name followed by a semicolon-separated list of numerical values. The sensor names include "simstorevent" and "simsensorevent", both with "TYPE\_ACCELEROMETER" and a "30" or "29" parameter. The numerical values represent acceleration data in three dimensions. The terminal text is as follows:

```
simstorevent;0;TYPE_ACCELEROMETER;30;-7.269268;6.442955;-1.2984961
simsensorevent;0
;TYPE_ACCELEROMETER;29;-7.802751;5.912181;-0.4507785
simsensorevent;0;TYPE_ACCELE
ROMETER;29;-7.9960017;5.662624;-0.19668841
simsensorevent;0;TYPE_ACCELEROMETER;30
;-7.7037888;5.9877224;-0.9158656
simsensorevent;0;TYPE_ACCELEROMETER;29;-7.365481
4;6.25211;-1.6434166
simsensorevent;0;TYPE_ACCELEROMETER;29;-6.9947886;6.4315786;
-2.397442
simsensorevent;0;TYPE_ACCELEROMETER;29;-7.32052;6.1887913;-2.036875
simstorevent;0;TYPE_ACCELEROMETER;29;-7.9070582;5.667595;-1.1818664
simsensorevent;
0;TYPE_ACCELEROMETER;30;-8.398877;5.0363092;-0.36670408
simsensorevent;0;TYPE_ACC
ELEROMETER;29;-8.781941;4.333909;0.36712438
simsensorevent;0;TYPE_ACCELEROMETER;2
9;-8.588058;4.710294;-0.31367403
simsensorevent;0;TYPE_ACCELEROMETER;29;-8.348017
;5.032569;-1.0118593
```

# Example of analysing each simulation

timestamp	s <sub>1</sub>	s <sub>2</sub>	s <sub>3</sub>	s <sub>4</sub>	s <sub>5</sub>	s <sub>6</sub>	known behaviour	predicted Algorithm 1	predicted Algorithm 2
1514819514501	0	0	0	1	0	0	GoToKitchen10	Normal	Disoriented
....									
1514822418501	0	0	0	1	0	0	GoGetLost15	Normal	Disoriented
1514822434501	0	1	0	0	0	0	GoGetLost15	Normal	Normal
....									
1514822447501	0	0	0	0	0	1	GoGetLost15	Normal	Normal
1514822478501	0	0	0	1	0	0	GoGetLost15	Normal	Disoriented
1514822493501	0	1	0	0	0	0	GoGetLost15	Disoriented	Normal
....									
1514822507501	0	0	0	0	0	1	GoGetLost15	Disoriented	Normal
1514822538501	0	0	0	1	0	0	GoGetLost15	Disoriented	Disoriented
....									

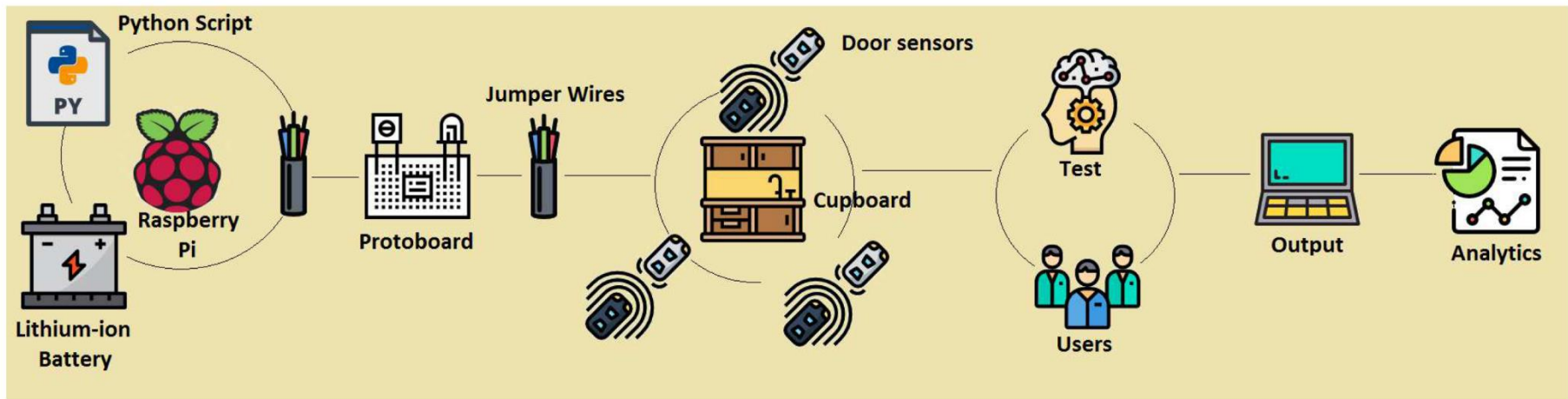
# Results of the analysis of simulated navigation patterns

	<b>Algorithm 1</b>	<b>Algorithm 2</b>
<b>Accuracy</b>	90.72%	87.63%
<b>Precision</b>	83.33%	60.00%

# A case Study: A smart cupboard

- International collaboration between:
  - University Complutense of Madrid, Madrid, Spain
  - Edison Desarrollos Company, Teruel, Spain
  - **Harvard University**, Boston, United States of America
  - Massachusetts General Hospital, United States of America
  - University of Zaragoza, Teruel, Spain
- Publication in international journal with impact:
  - González-Landero, F., García-Magariño, I., Amariglio, R., & Lacuesta, R. (2019). [Smart Cupboard for Assessing Memory in Home Environment](#). *Sensors*, 19(11), 2552.
- Available in TeleMadrid television, RCN Radio Colombia and more than 10 newspapers.

# Overview of the Smart Cupboard approach

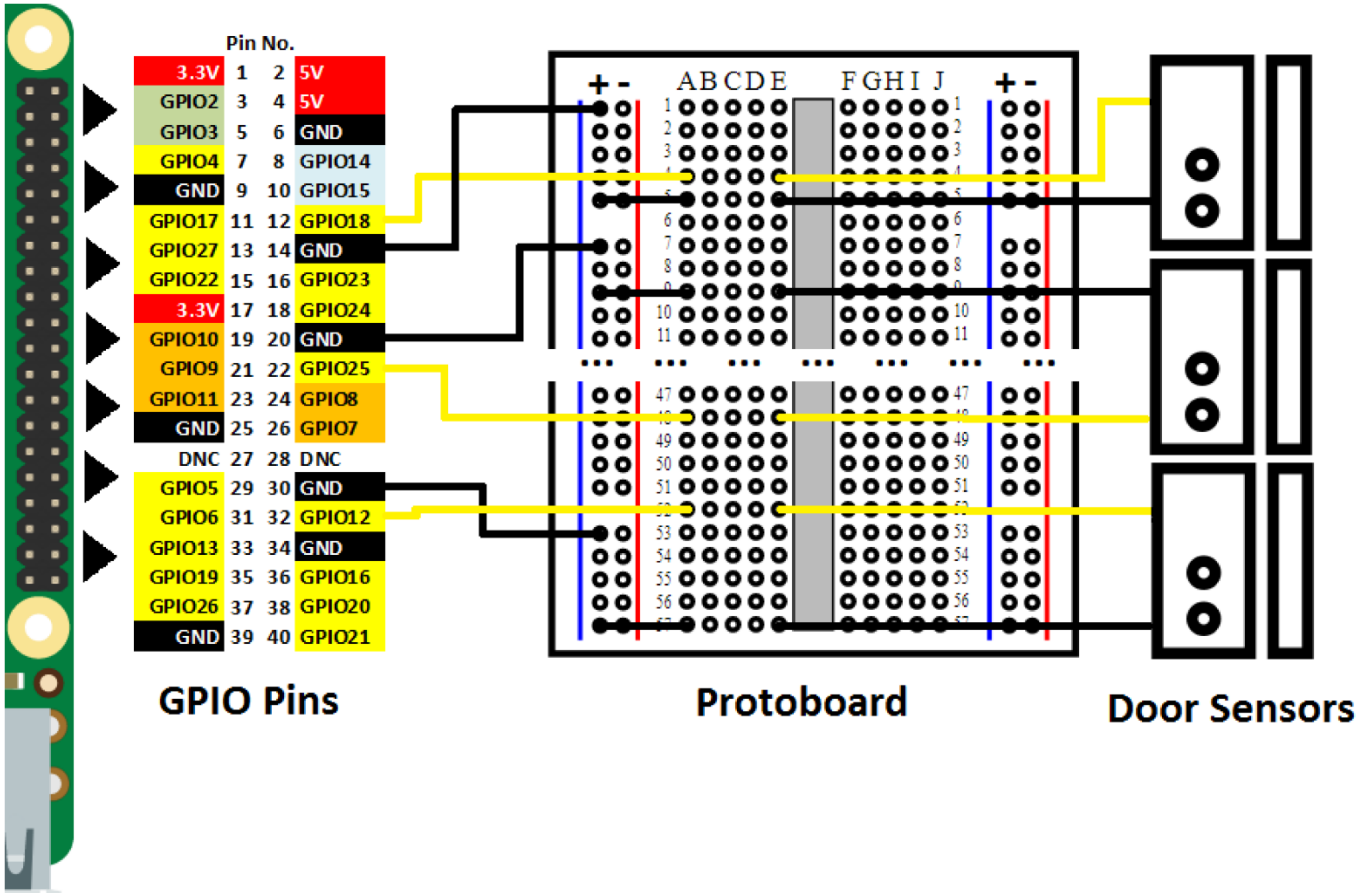




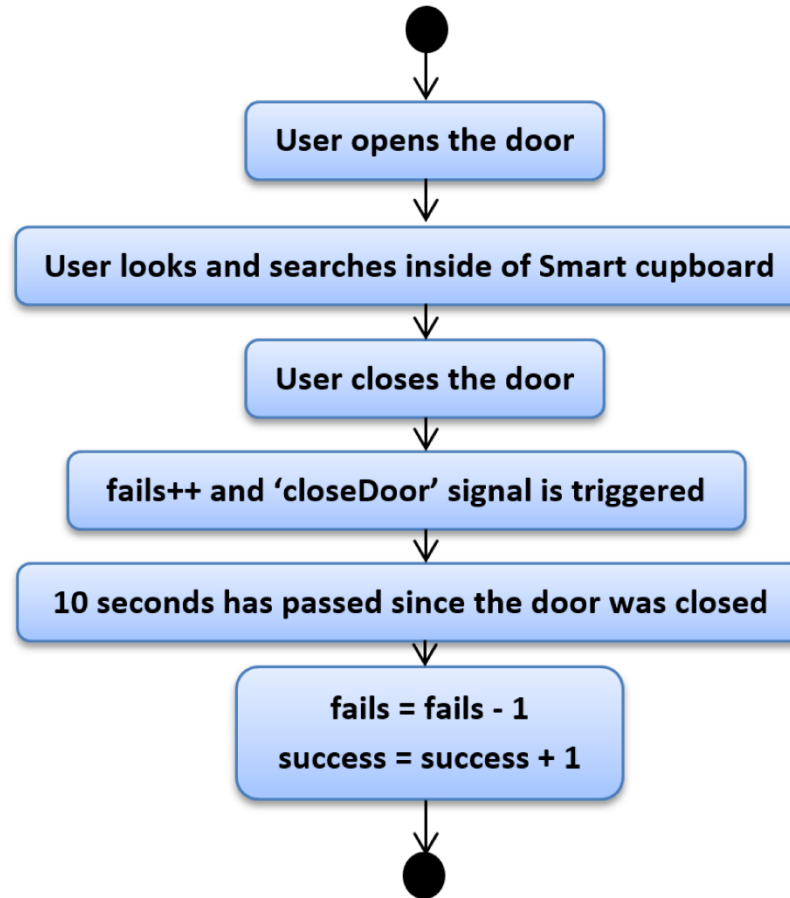
# Magnetic Door Sensor



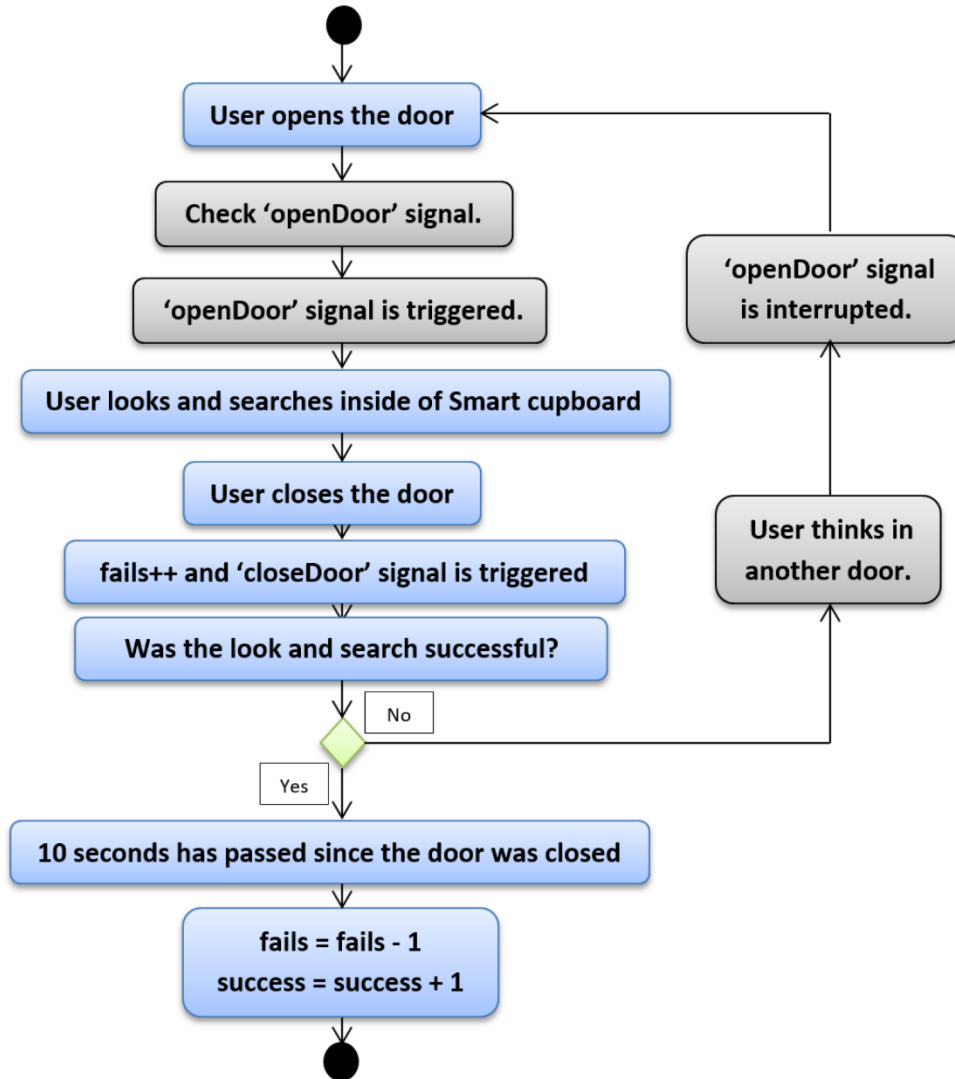
# Schema of the Connections



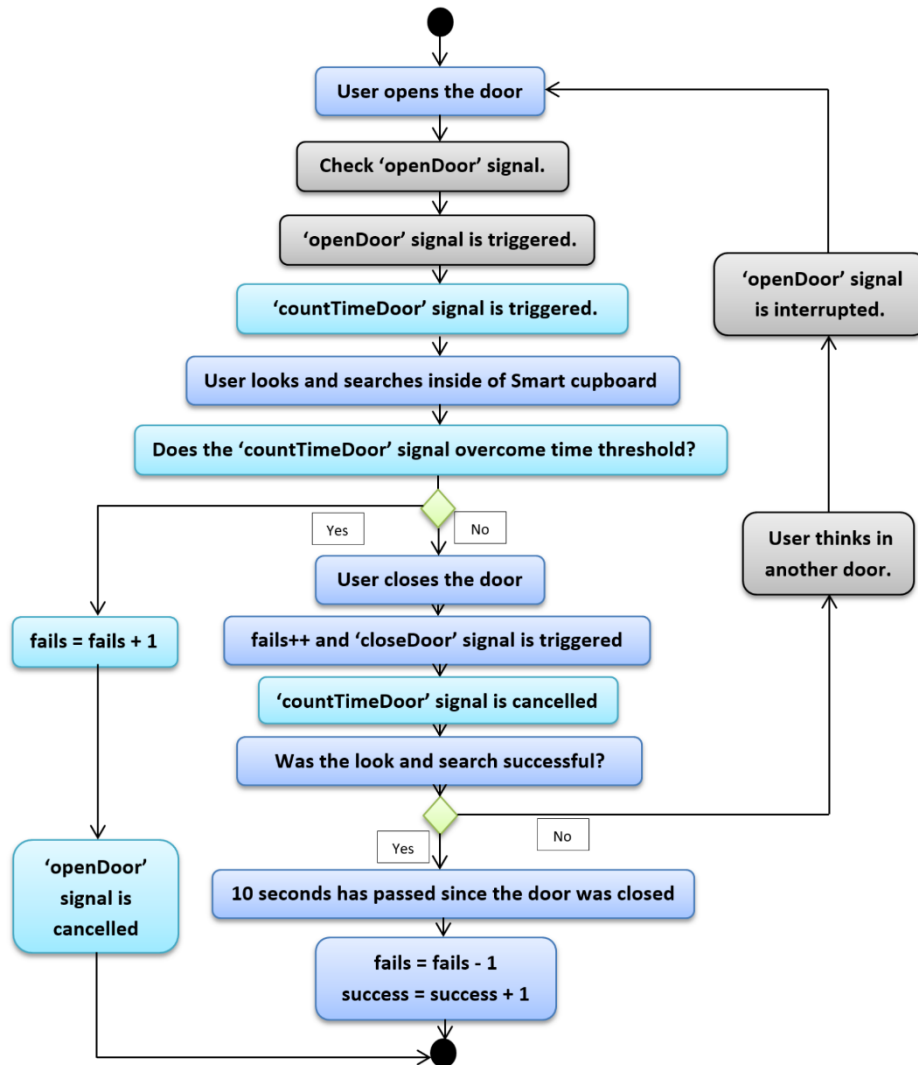
# Normal Use of Smart Cupboard: Sequence of Actions



# User Forgets the Location of something- Detection



# User keeps looking inside a compartment for a long time



# Implementation Details

```
01. import RPi.GPIO as GPIO
02.
03. #Set Broadcom mode so we can address GPIO pins by number
04. GPIO.setmode(GPIO.BCM)
05.
06. #This is the GPIO pin number we have one of the door sensor
07. #wires attached to, the other should be attached to a ground
08. DOOR_SENSOR_PIN_ONE = 18
09. DOOR_SENSOR_PIN_TWO = 12
10. DOOR_SENSOR_PIN_THREE = 25
11.
12. #Set up the door sensor pin
13. GPIO.setup(DOOR_SENSOR_PIN_ONE, GPIO.IN, pull_up_down = GPIO.PUD_UP)
14. GPIO.setup(DOOR_SENSOR_PIN_TWO, GPIO.IN, pull_up_down = GPIO.PUD_UP)
15. GPIO.setup(DOOR_SENSOR_PIN_THREE, GPIO.IN, pull_up_down = GPIO.PUD_UP)
16.
17. while True:
18.     oldIsOpenOne = isOpenOne
19.     isOpenOne = GPIO.input(DOOR_SENSOR_PIN_ONE)
20.
21.     oldIsOpenTwo = isOpenTwo
22.     isOpenTwo = GPIO.input(DOOR_SENSOR_PIN_TWO)
23.
24.     oldIsOpenThree = isOpenThree
25.     isOpenThree = GPIO.input(DOOR_SENSOR_PIN_THREE)
```

# Smart Cupboard Assembled



# Experimentation

## Order of Objects in the Experimentation (random to avoid reasoning for retrieving location)

Object	Compartment	Round
Cup Sweet Corn Chili Egg Box of Matches	First	First
Evaporated Milk Soda Breadcrumb Beer Chili peppers	Second	
Potato Lentils Olives Mayonnaise Chocolate milkshake	Third	

Object	Compartment	Round
Grapes Soup cube Peach in syrup Condensed milk Salt	First	Second
Baking powder Green peas Bread of milk Jam teaspoon	Second	
Sausages Honey Tuna Tea Oregano	Third	






# Example of Distribution of Food for Experiments



# Control Test

## Test of Face-Name pairs (well-known and validated in the literature)

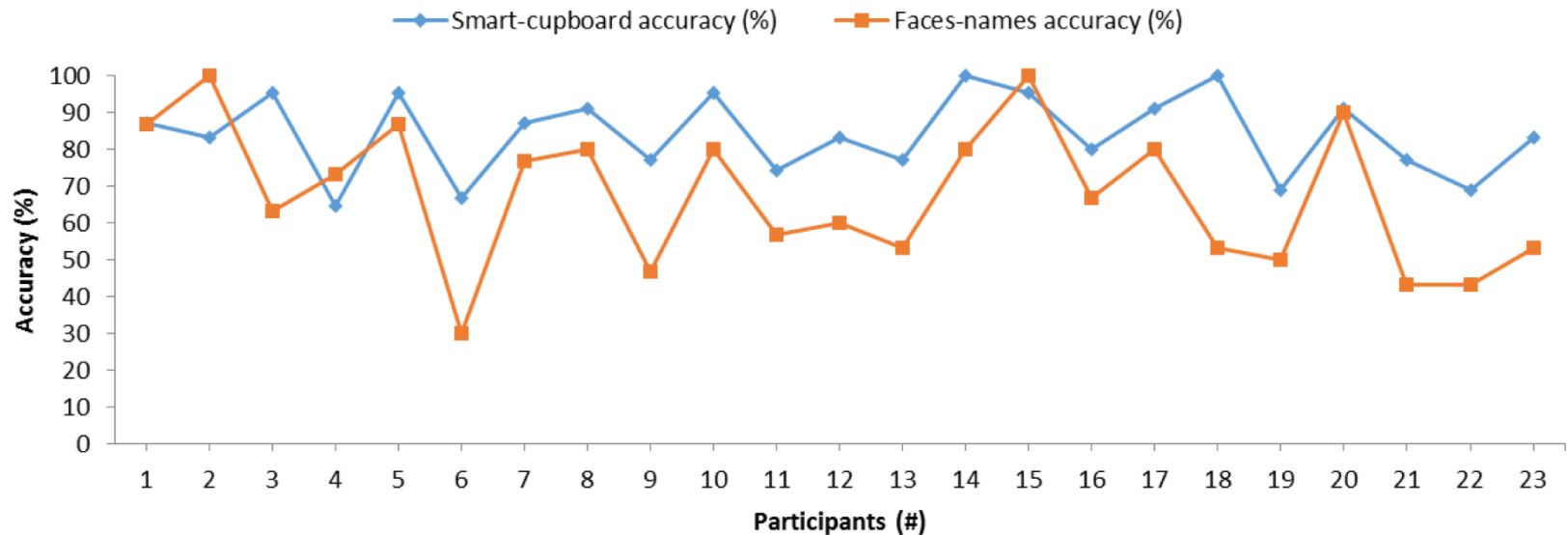
<p>Question 1</p>  <p>a) Elena b) Bethania c) Inés d) Juana</p>	<p>Question 2</p>  <p>a) Ester b) Meritxell c) Amalia d) Agustina</p>	<p>Question 3</p>  <p>a) Hugo b) Cirino c) Xesús d) Vidal</p>
--	--	--

## Self-reported test

- Do you have difficulty in remembering people's names or phone numbers?
- How often do you find yourself trying to remember the location of everyday items (e.g., your keys, wallet, glasses, etc.)?
- How often do you have to replace passwords (numerical or verbal) because you've forgotten the original one?
- How often do you find yourself asking questions like, "What was I about to do next?"

...

# Comparison of Accuracies of Smart cupboard and Face-names



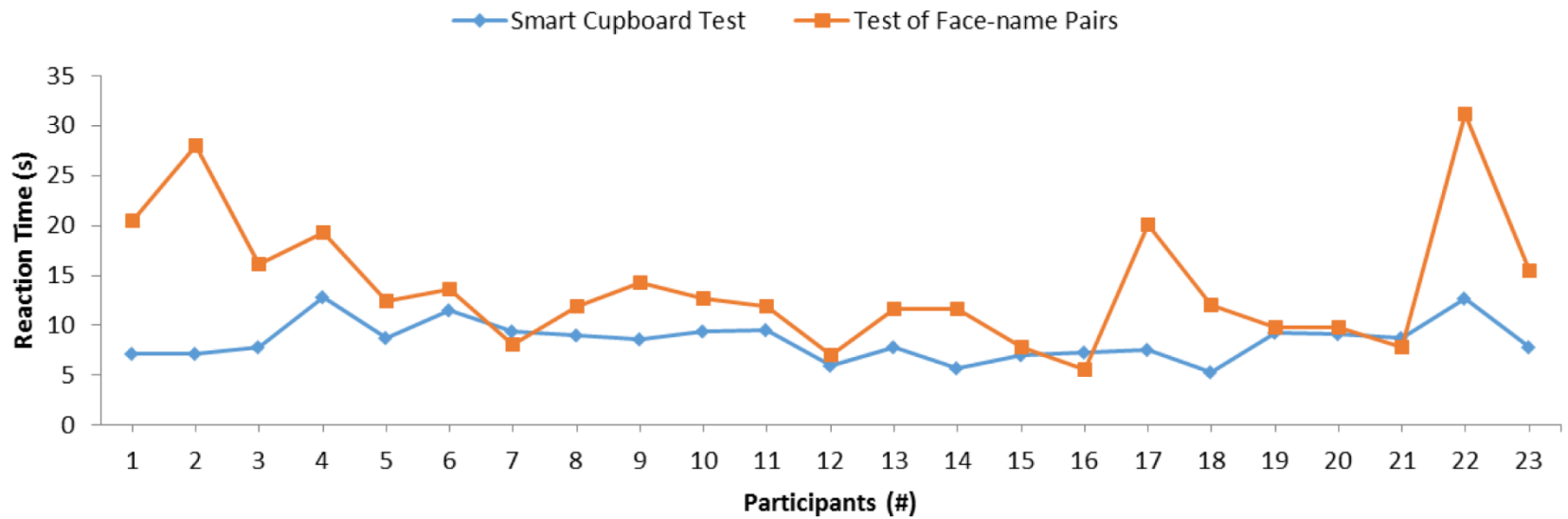
# Comparison of Accuracies of Smart cupboard and Face-names

		Accuracy Smart Cupboard	Faces-Name Test
Accuracy Smart Cupboard	Pearson Correlation	1	.597**
	Sig. (2-tailed)		.003
	N	23	23
Faces-Name Test	Pearson Correlation	.597**	1
	Sig. (2-tailed)	.003	
	N	23	23

\*\* . Correlation is significant at the 0.01 level (2-tailed).

Correlation between the accuracy of SC and the accuracy of Face-names test

# Comparison of Reaction Times

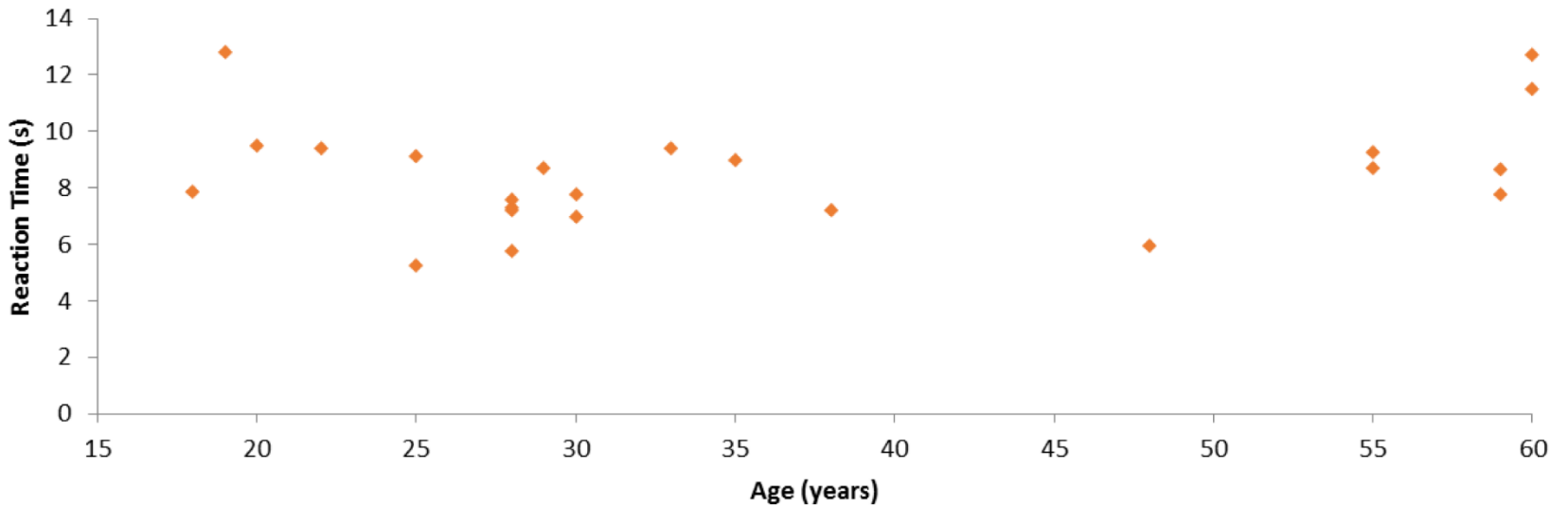


# Comparison of Reaction Times

		Reaction Time Smart Cupboard	Reaction Time Face-Name Test
Reaction Time Smart Cupboard	Pearson Correlation	1	.341
	Sig. (2-tailed)		.111
	N	23	23
Reaction Time Face-Name Test	Pearson Correlation	.341	1
	Sig. (2-tailed)	.111	
	N	23	23

Correlation between the reaction time of SC and the reaction time of Face-names test

# Reaction time in smart cupboard and Age



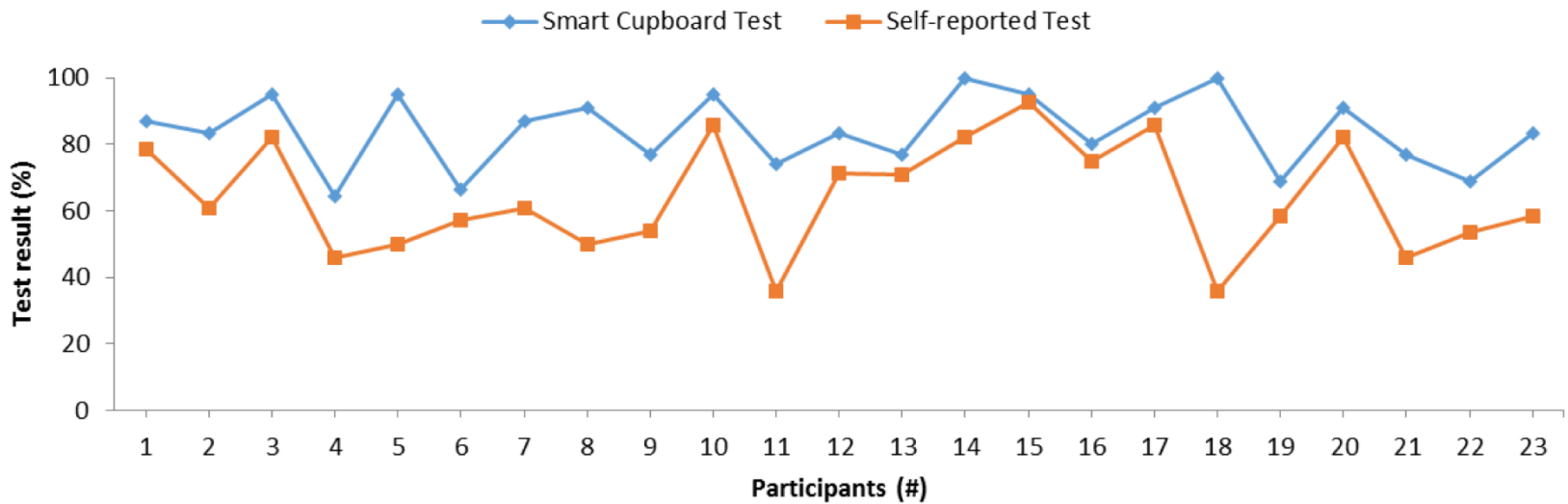
# Reaction time in smart cupboard and Age

		Age	Reaction Time Smart Cupboard
Age	Pearson Correlation	1	.092
	Sig. (2-tailed)		.699
	N	20	20
Reaction Time Smart Cupboard	Pearson Correlation	.092	1
	Sig. (2-tailed)	.699	
	N	20	20

Correlation between the reaction time and age of participants in the smart cupboard test



# Comparison of Smart Cupboard and Self-Reported Test



# Comparison of Smart Cupboard and Self-Reported Test

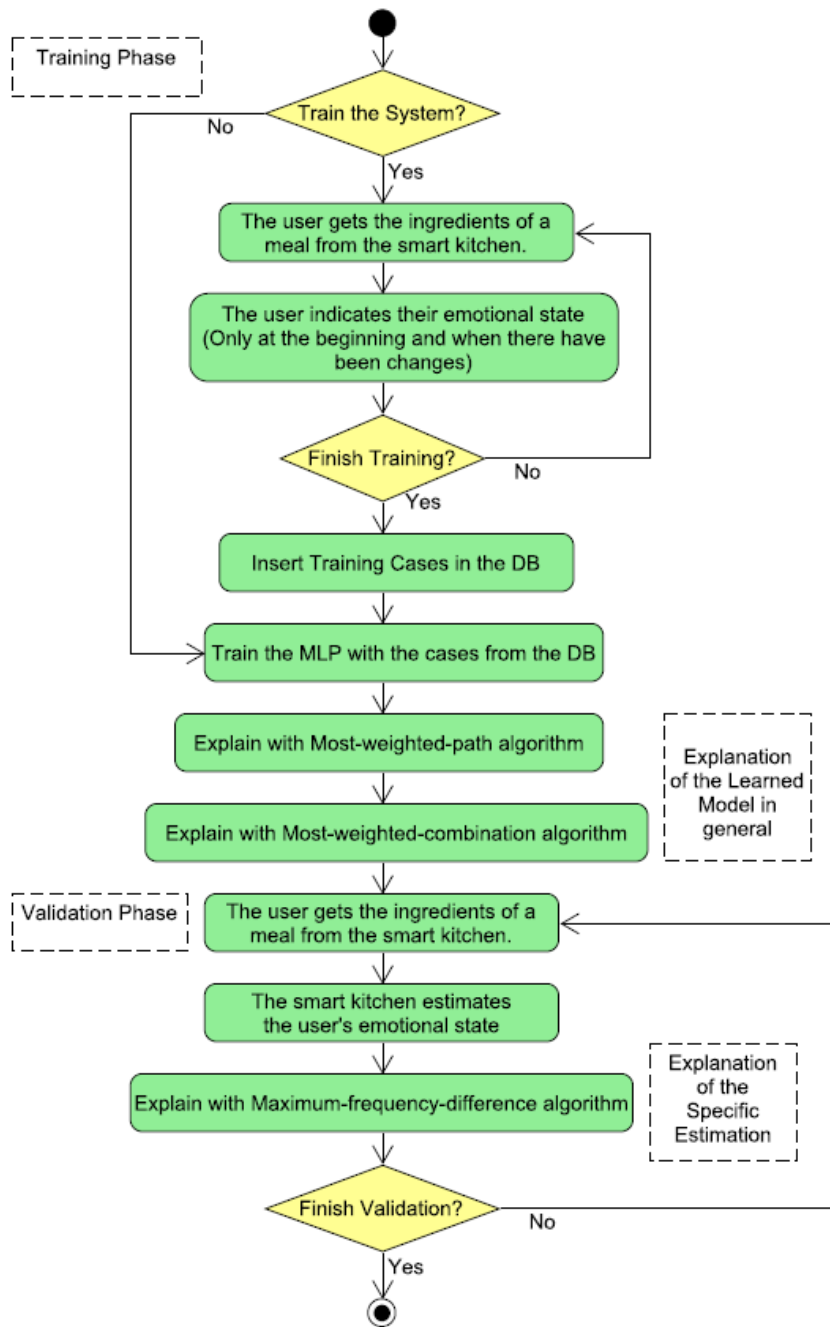
		Smart Cupboard	Accuracy Self-Reported test
Smart Cupboard	Pearson Correlation	1	.443 <sup>*</sup>
	Sig. (2-tailed)		.034
	N	23	23
Accuracy Self-Reported test	Pearson Correlation	.443 <sup>*</sup>	1
	Sig. (2-tailed)	.034	
	N	23	23

\*. Correlation is significant at the 0.05 level (2-tailed).

Correlation between accuracy of SC and accuracy of Self-reported test

# Human-Centric Artificial Intelligence in smart kitchens

- Reference:
  - García-Magariño, I., Muttukrishnan, R., & Lloret, J. (2019). [Human-centric AI for trustworthy IoT systems with explainable multilayer perceptrons](#). IEEE Access, 7 (1), 125562-125574
- International Collaboration
  - University Complutense of Madrid, Spain
  - **City, University of London**, United Kingdom
  - Polytechnical University of Valencia, Spain
- Concept:
  - Auto-generate easy-to-understand explanations



Approach for experimenting HAI algorithms in simulated smart kitchens

---

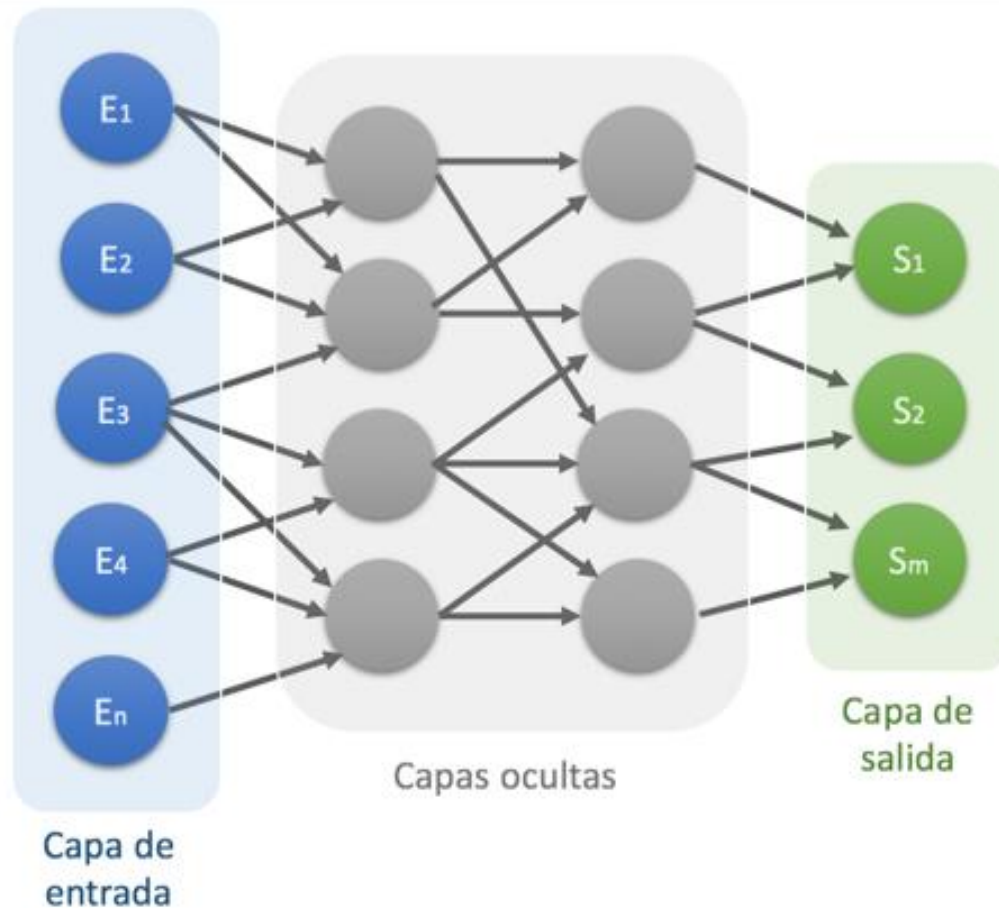
**Algorithm 1 Most-Weighted-Path Explanation:** It Provides a HAI Explanation Based on the Path From the Output to the Most Relevant Input Based on the Selection of the Most Weighted Dendrites

---

```
1: function explainMostWeightedPath(mlp, names)
2:   current ← mlp.outputNeuron
3:   while mlp.isNeuron(current) do
4:     dendrite ← mostWeightedDendrite(neuron)
5:     current ← mlp.connectedTo(dendrite)
6:   input ← current
7:   inputName ← names.inputs[input]
8:   explanation ← ‘In the learned model for the
   ‘+names.IoTsystem+, the most relevant input for esti-
   mating that you are ‘+ names.highestOutputValue+’ is
   that ‘+names.userAction+’ ‘+ inputName+’.’
9:   return explanation
```

---

# Esquema de Red Neuronal MLP



**Algorithm 2 Most-Weighted-Combination Explanation:**

It Provides a HAI Explanation Based on the Most Relevant Combination of Two Inputs Based the Most Weighted Dendrites of the First and Second Neuron Layers

---

```
1: function explainMostWeightedCombination(mlp,
    names)
2:   layer  $\leftarrow$  1  $\triangleright$  Second layer, as count starts on 0
3:   numInputs  $\leftarrow$  2
4:   dendrites  $\leftarrow$  mlp.getDendritesLayer(layer)
5:   quicksortByWeightDescendentOrder(dendrites)
6:   found  $\leftarrow$  false
7:   i  $\leftarrow$  0
8:   while i < dendrites.length and not found do
9:     dendrite  $\leftarrow$  dendrites[i]
10:    inputNeuron  $\leftarrow$  mlp.connectedTo(dendrite)
11:    inputDendrites  $\leftarrow$  mlp.mostWeightedDendrites(
12:      neuron, numInputs)
13:    found  $\leftarrow$  true
14:    for j  $\in$  [0, numInputs) do
15:      if inputDendrites[j].weight > threshold then
16:        combination[j]  $\leftarrow$  mlp.connectedTo(
17:          inputDendrites[j])
18:      else
19:        found  $\leftarrow$  false
20:    i  $\leftarrow$  i + 1
21:    if found then
22:      explanation  $\leftarrow$  'In the learned model
    for the '+names.IoTsystem+', the most
    relevant input combination for estimating
    whether you are '+names.highestOutputValue+'
    is that '+names.userAction+' and
    '+names.inputs[combination[0]]+' and
    '+names.inputs[combination[1]]+'.'
23:    else
24:      explanation  $\leftarrow$  'No combination of two inputs is
    especially relevant.'
25:    return explanation
```

---

---

**Algorithm 5 Maximum-Frequency-Difference Explanation:**

It Provides a HAI Explanation Based on the Most Discriminative Input, Measured as the One With the Highest Difference of Frequency Percentage for the Given Prediction

---

```
1: function explainMaxFreqDiff(caseInputs, prediction,
   names)
2:   maxDiff  $\leftarrow$  minIntValue
3:   maxInputName  $\leftarrow$  ''
4:   for i  $\in$  [0,names.inputs.length) do
5:     if caseInputs[i] then
6:       inputName  $\leftarrow$  names.inputs[i]
7:       diff  $\leftarrow$  DiffPercen(inputName,prediction)
8:       if diff > maxDiff then
9:         maxInputName  $\leftarrow$  inputName
10:        maxDiff  $\leftarrow$  diff
11:   explanation  $\leftarrow$  'The '+names.IoTsystem+' estimates that you are '+prediction+' because among other reasons '+names.userAction+' '+maxInputName+', which is '+maxDiff+'% more frequent in people in this '+names.state+' than in people with other '+names.states+'.'
12:   return explanation
```

---



# Smart Kitchen Simulator (training phase)

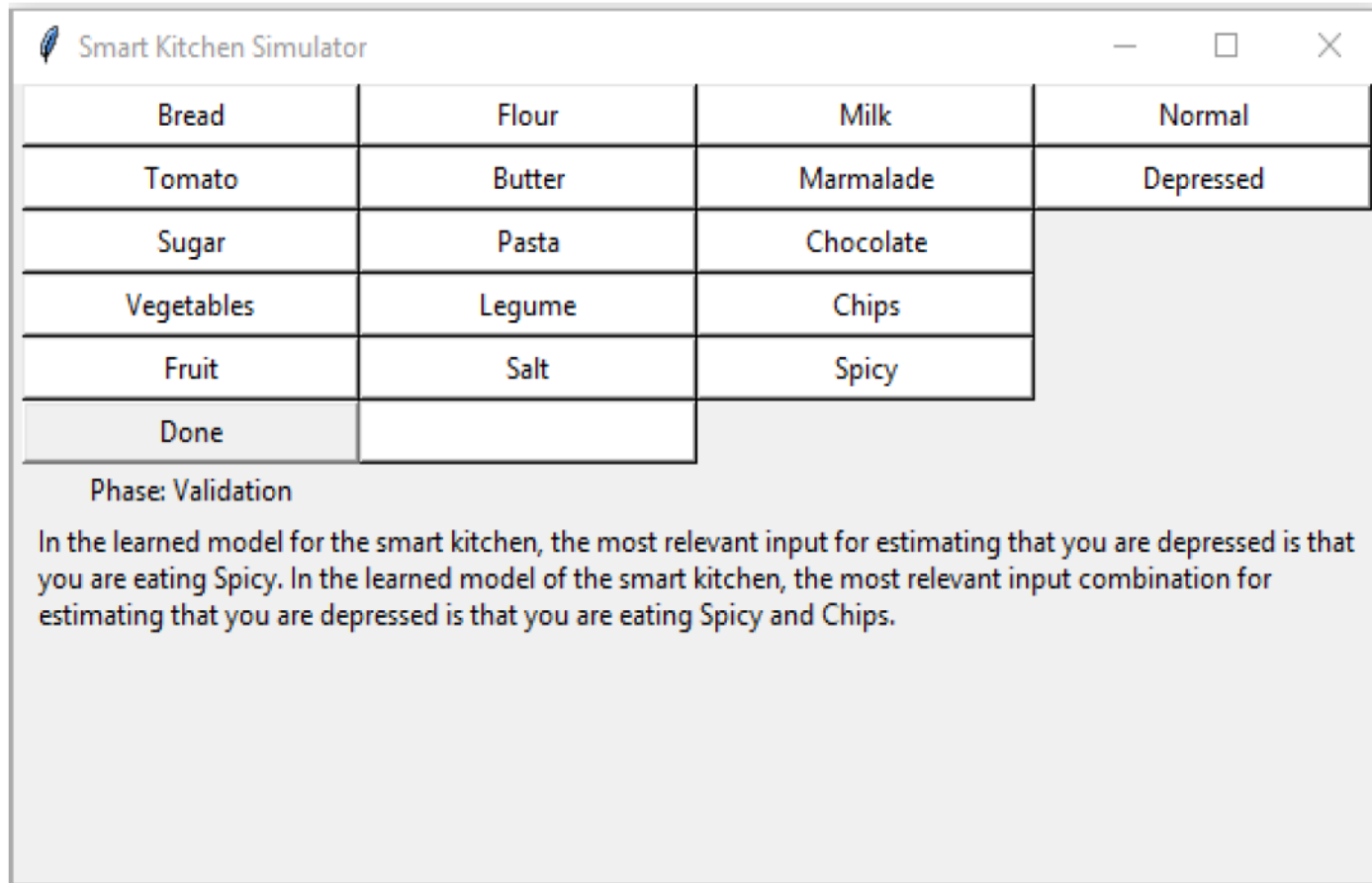


The image shows a window titled "Smart Kitchen Simulator" with standard window controls (minimize, maximize, close). Inside the window is a table with 4 columns and 7 rows. The first three columns contain ingredients, and the fourth column contains a status or mood. The first two rows have alternating pink and white background colors. The last row has a grey background. Below the table, the text "Phase: Training" is displayed.

Bread	Flour	Milk	Normal
Tomato	Butter	Marmalade	Depressed
Sugar	Pasta	Chocolate	
Vegetables	Legume	Chips	
Fruit	Salt	Spicy	
Done	Finish Training		

Phase: Training

# Auto-generated explanation for the learned model



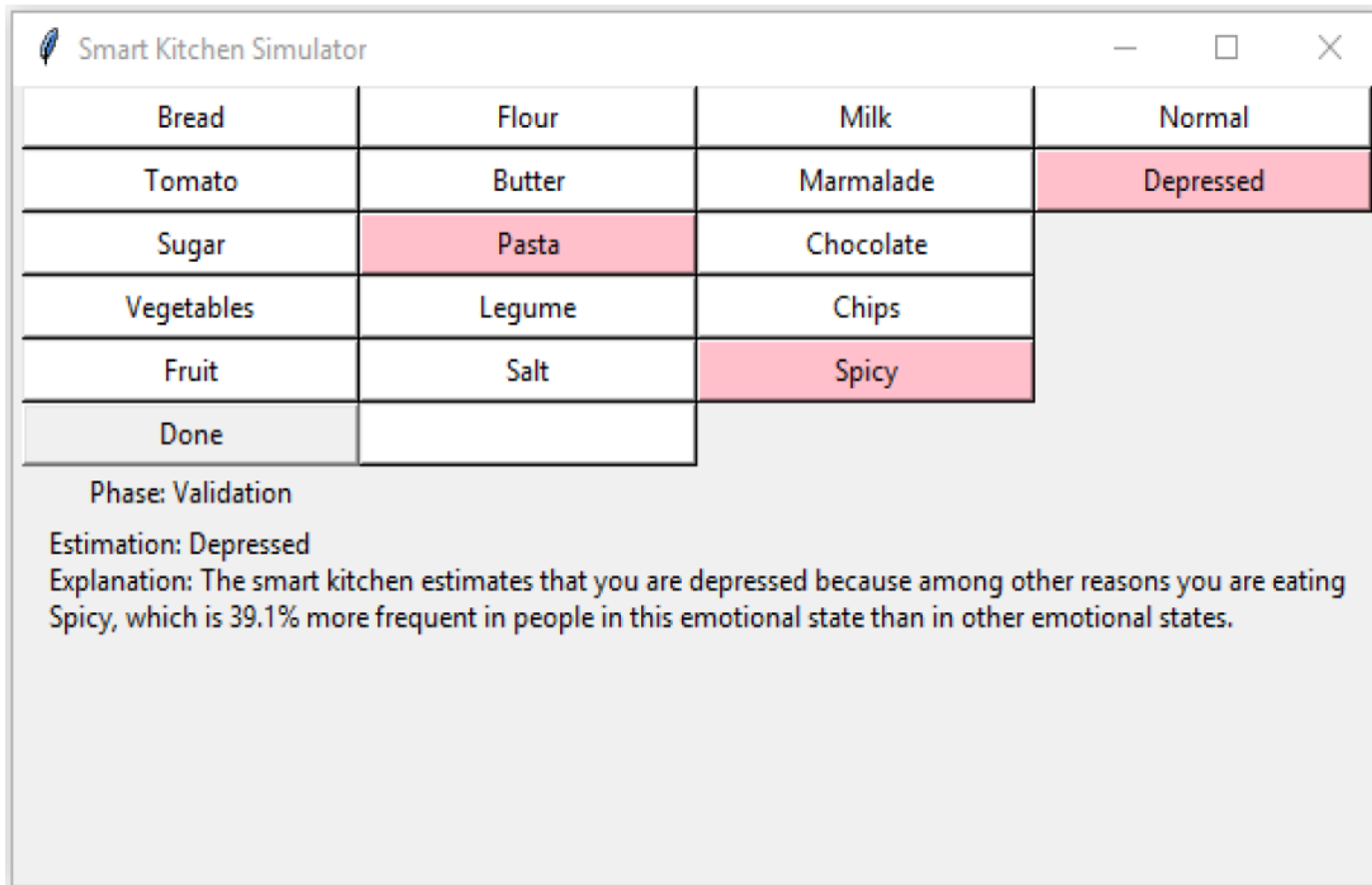
Smart Kitchen Simulator

Bread	Flour	Milk	Normal
Tomato	Butter	Marmalade	Depressed
Sugar	Pasta	Chocolate	
Vegetables	Legume	Chips	
Fruit	Salt	Spicy	
Done			

Phase: Validation

In the learned model for the smart kitchen, the most relevant input for estimating that you are depressed is that you are eating Spicy. In the learned model of the smart kitchen, the most relevant input combination for estimating that you are depressed is that you are eating Spicy and Chips.

# Auto-generated explanation for one meal



Smart Kitchen Simulator

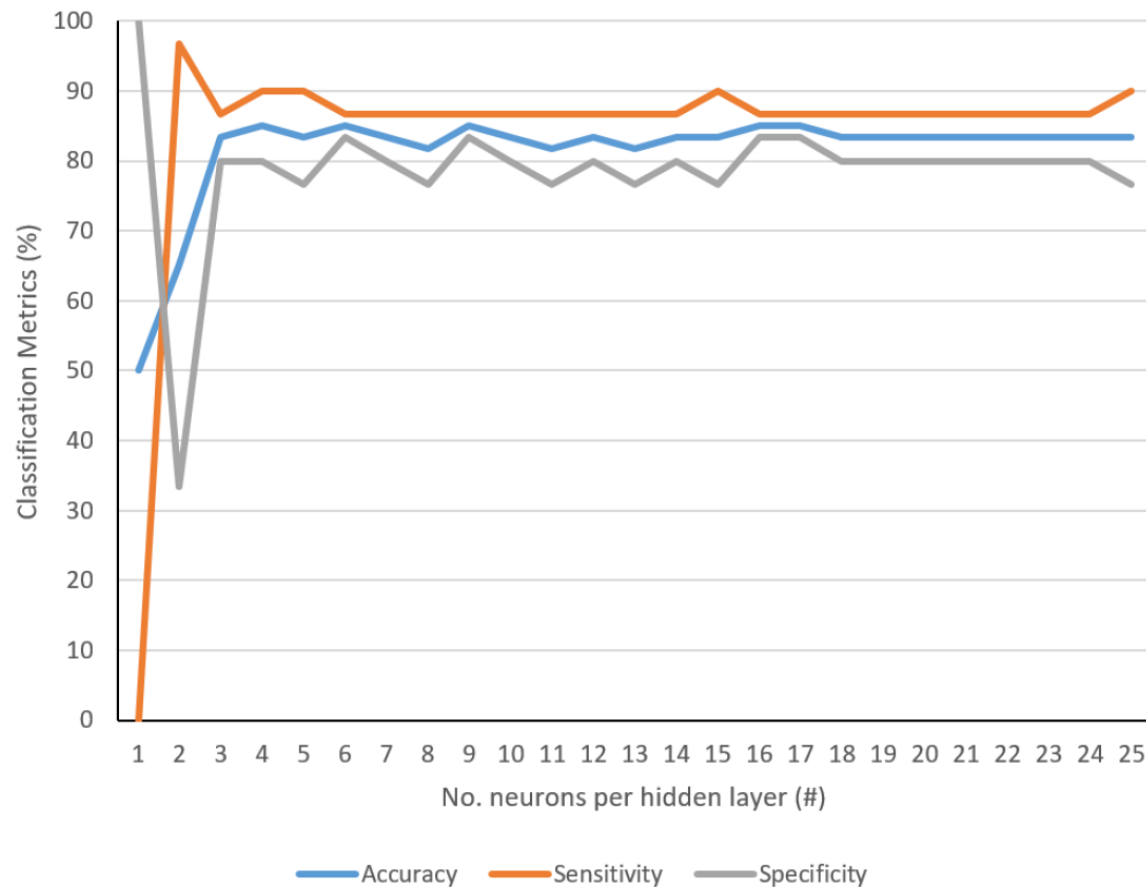
Bread	Flour	Milk	Normal
Tomato	Butter	Marmalade	Depressed
Sugar	Pasta	Chocolate	
Vegetables	Legume	Chips	
Fruit	Salt	Spicy	
Done			

Phase: Validation

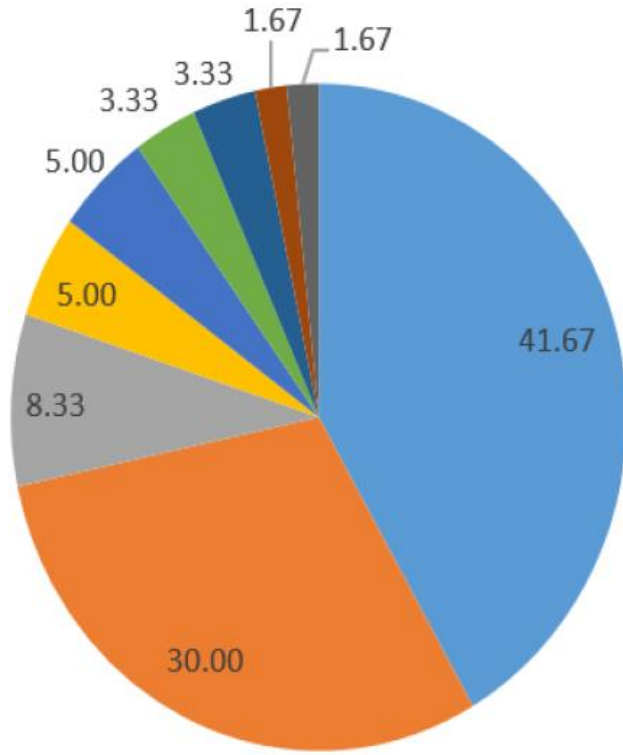
Estimation: Depressed

Explanation: The smart kitchen estimates that you are depressed because among other reasons you are eating Spicy, which is 39.1% more frequent in people in this emotional state than in other emotional states.

# Adjusting Neural Network parameters



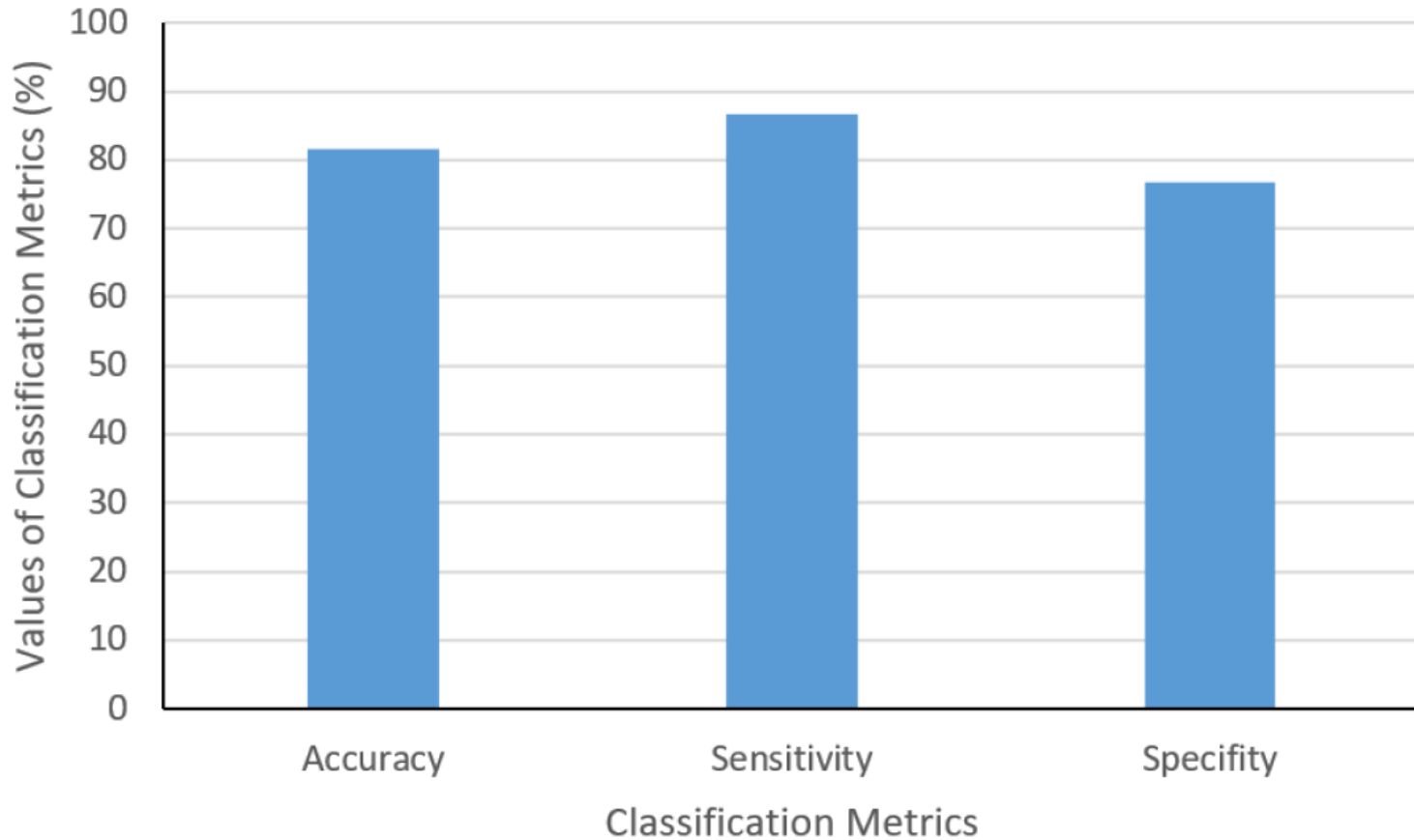
Frequency of Explanations (%)



# Results about HAI auto-generated explanations

- Reasonable Vegetable explanation
- Reasonable Sugar explanation
- Reasonable Spicy explanation
- Reasonable Bread explanation
- Reasonable Pasta explanation
- Reasonable Flour explanation
- Other Reasonable explanations
- Non-Reasonable Salt explanation
- Non-Reasonable Tomato explanation

# Classification Metrics Results



# Conclusions and Future Work

- IoT for tracking memory in Activities of Daily Life
- Simulated navigation patterns
- Real Smart-cupboards
- Human-centric artificial intelligence can be applied (e.g. with explainable multi-layer perceptrons)
- Future: Development of more techniques for measuring memory with IoT, E.g.:
  - Smart taps
  - Curtain sensors in doors