Practical Applications of Distributed IoT Systems

Iván García-Magariño
Department of Software Engineering and Artificial Intelligence
Complutense University of Madrid, Spain
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

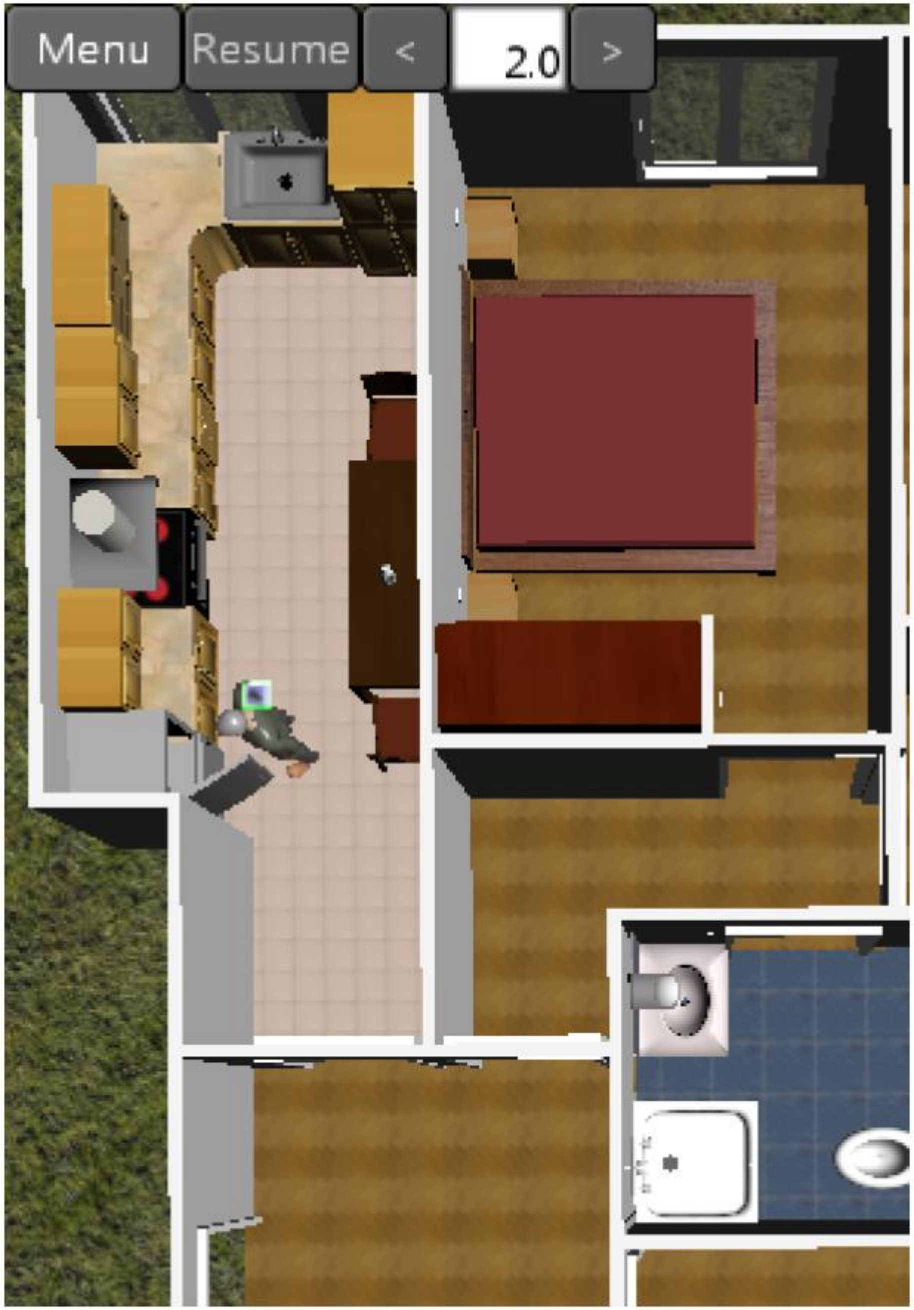
AIDE (developed by Grasia research group)

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

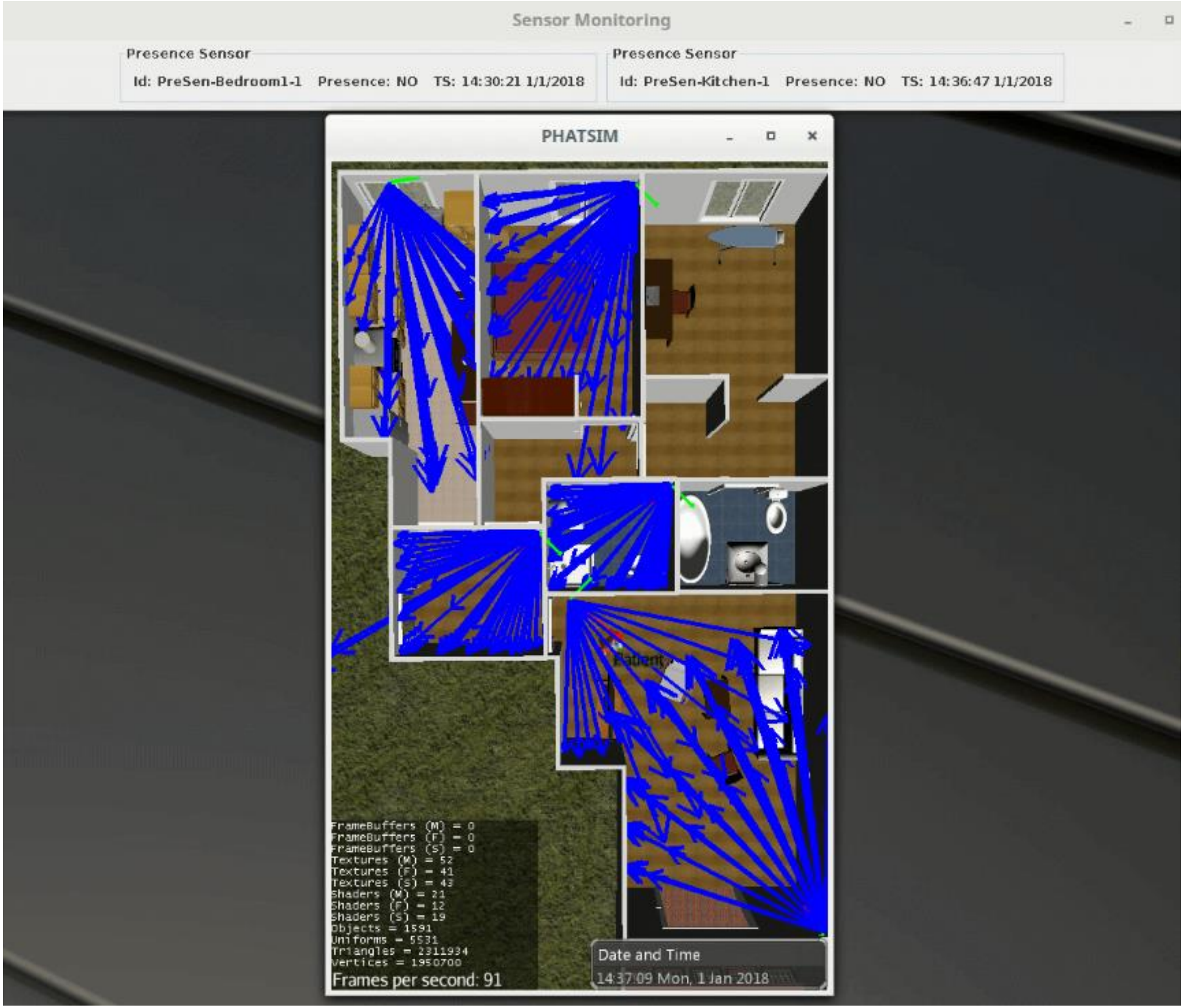
procedure InitializeMemoryTrackingSystem( )
    sum $\leftarrow$ 0
    queue $\leftarrow$ new CircularQueue()
procedure HandleSensorInformation($s_i, \Delta t_i, \Delta t_{s,i}$)
    queue.add (Δt_i)
    sum $\leftarrow$ sum + Δt_i
    if queue.length $> n_{p,1}$ **then**
        oldT $\leftarrow$ queue.begin
        queue.removeBegin()
        sum $\leftarrow$ sum - oldT
        if (sum $< t_{p,1}$) **then**
            notifyMemoryPattern (p_1)

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

procedure INITIALIZEMEMORYTRACKINGSYSTEM( )
    returningTimes $gets$ new List[$|S|$]
    for i $\in$ [0, $|S|$-1] **do**
        returningTimes[i] $\leftarrow$ new List();
procedure HANDLESENSORINFORMATION(s_i, Δt_i, $\Delta t_{s,i}$)
    returningTimes[s_i].add ($\Delta t_{s,i}$)
    if (returningTimes[s_i].length $> n_{p,2}$) and
(returningTimes[s_i].median $< t_{p,2}$) **then**
        notifyMemoryPattern (p_2)

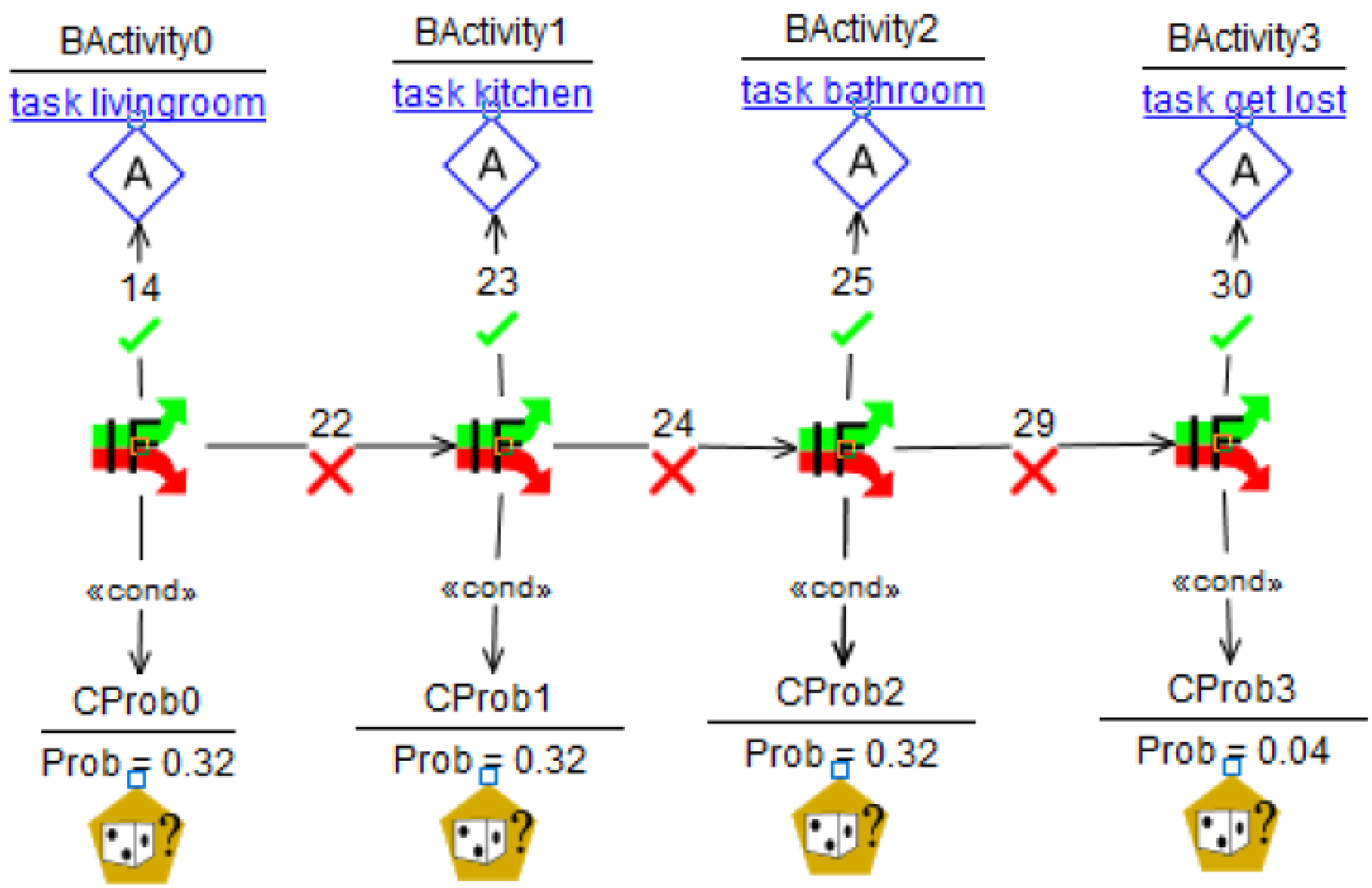


Example of the avatar opening the fridge

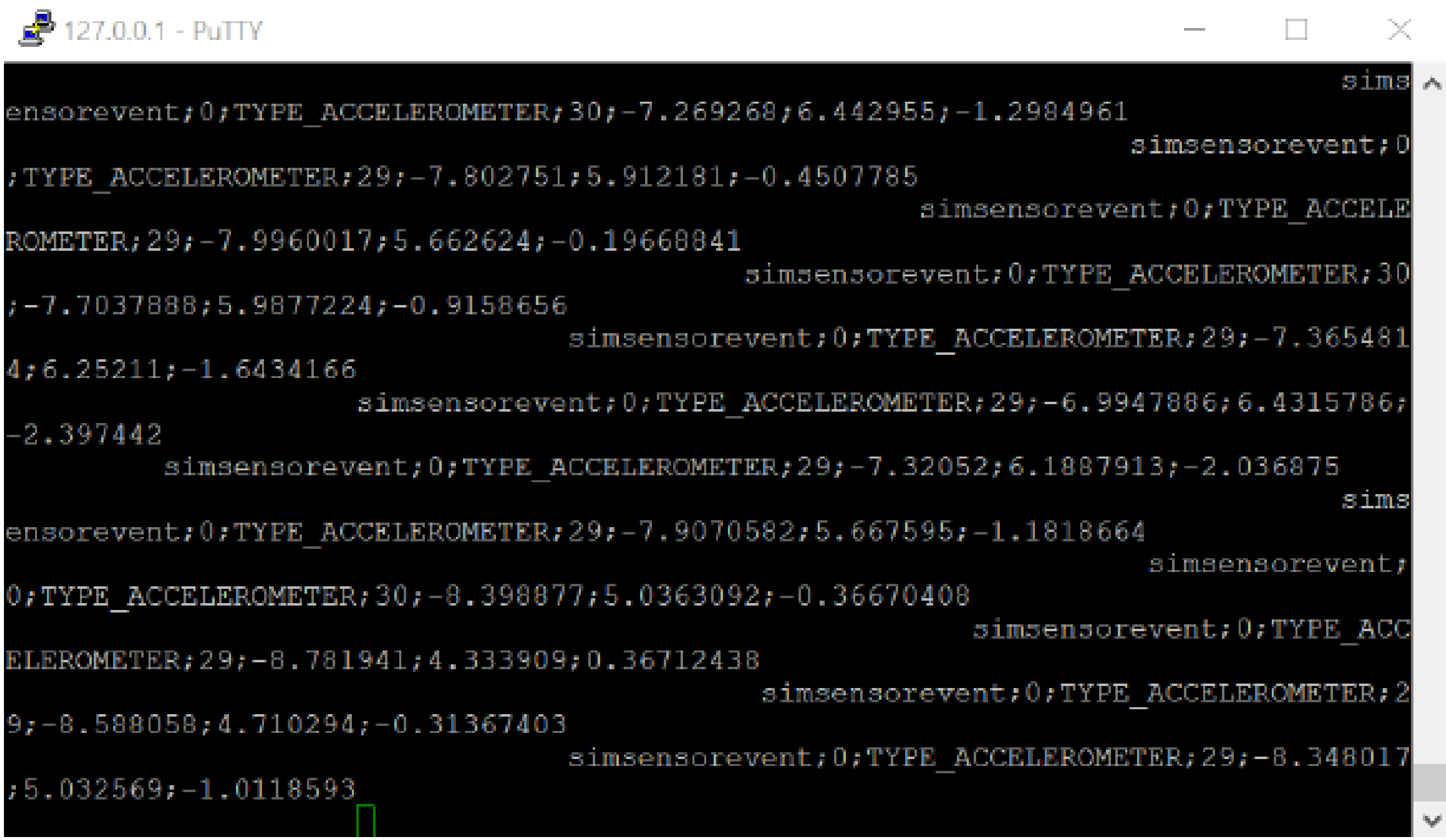


Presence Sensors in simulations

Modelling avatar behaviors



Example of information transmitted to ports in simulations



Example of analysing each simulation

timestamp	s_1	s_2	s_3	s_4	s_5	s_6	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
							....		

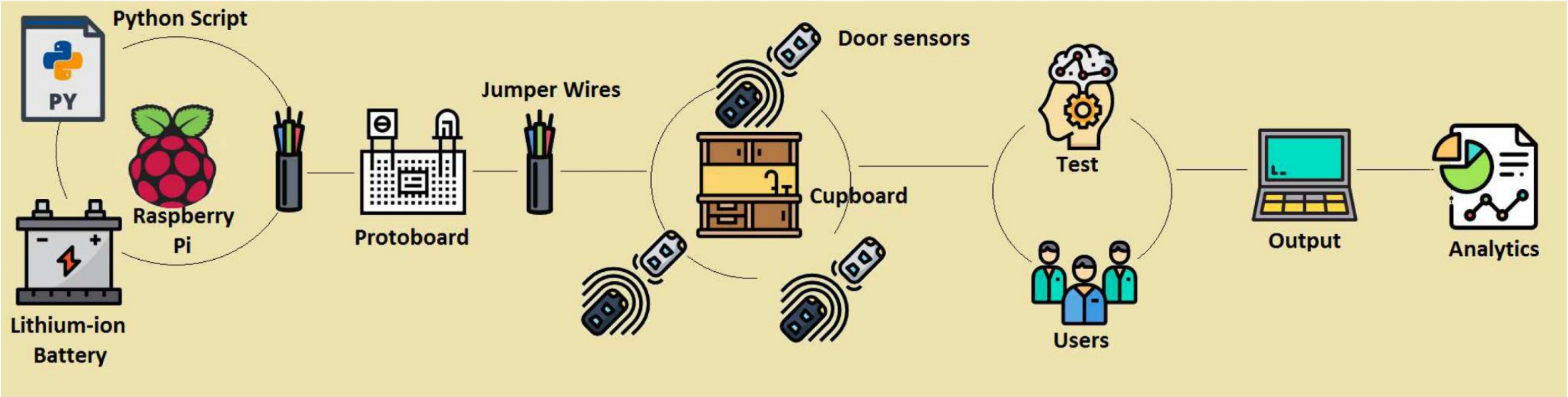
Results of the analysis of simulated navigation patterns

	Algorithm 1	Algorithm 2
Accuracy	90.72%	87.63%
Precision	83.33%	60.00%

A case Study: A smart cupboard

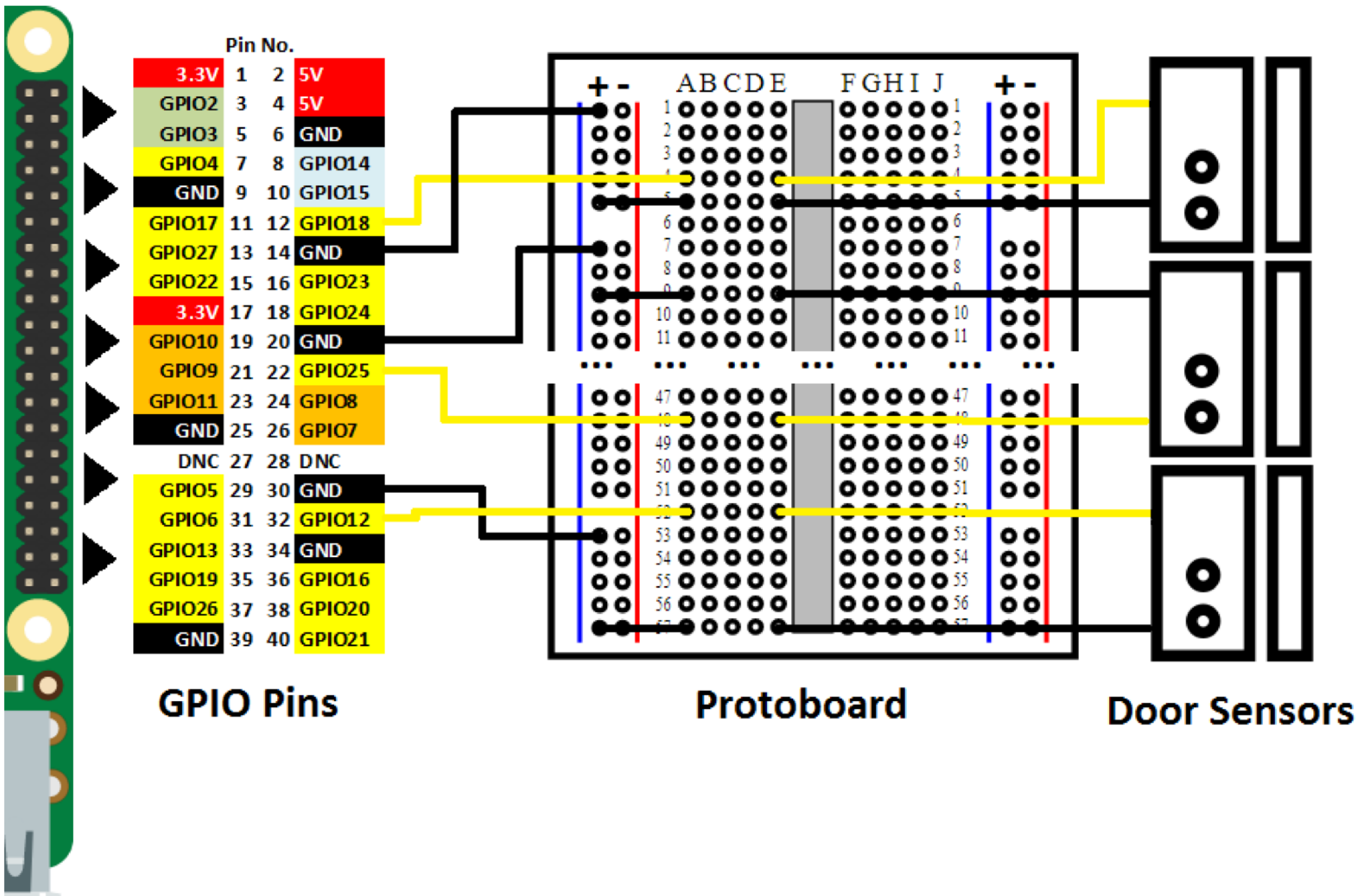
- International collaboration between:
  - University Complutense of Madrid, Madrid, Spain
  - Edison Desarrollos Company, Teruel, Spain
  - **Harvard University**, Boston, United States of America
  - Massachusetts General Hospital, United States of America
  - University of Zaragoza, Teruel, Spain
- 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

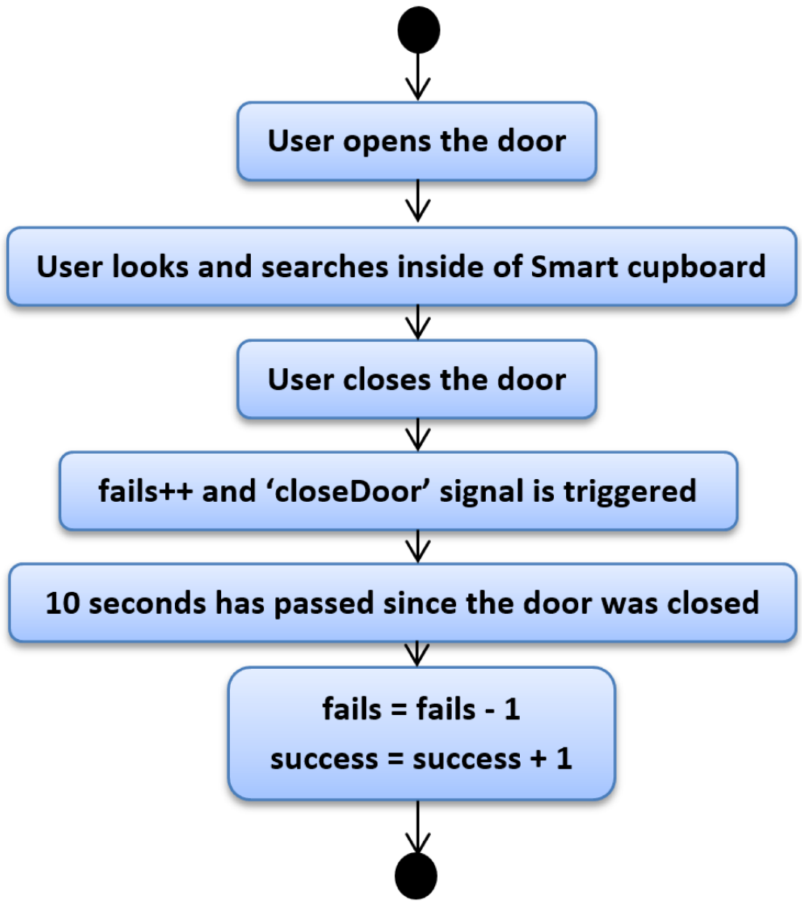


Magnetic Door Sensor

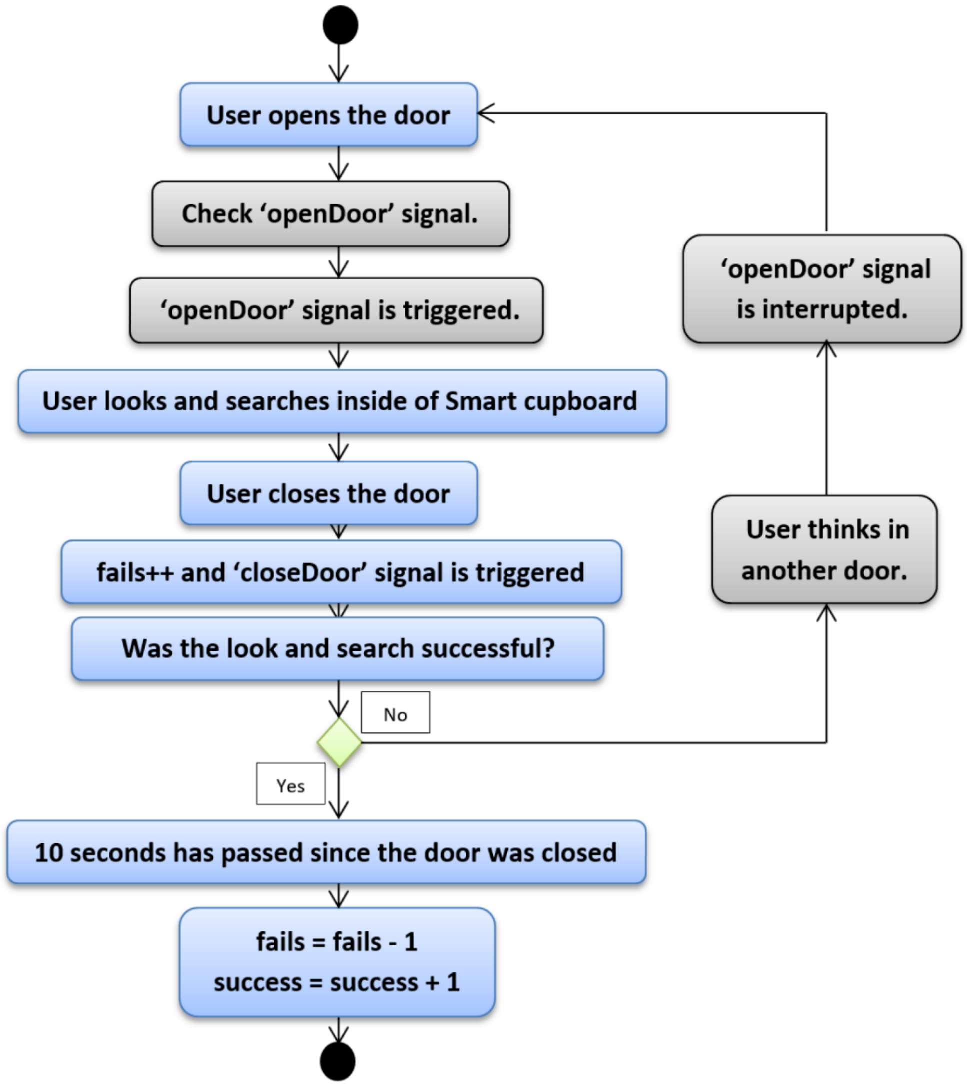
Schema of the Connections



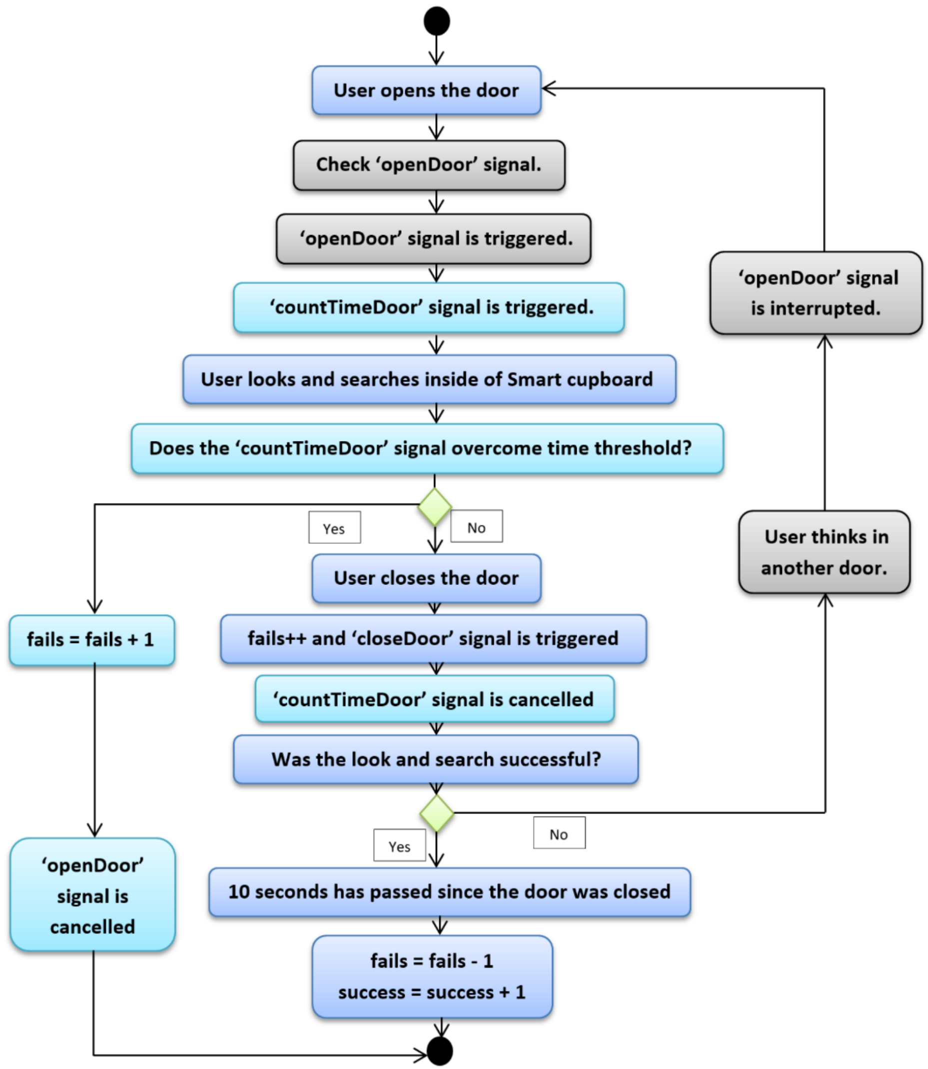
Normal Use of Smart Cupboard: Sequence of Actions



User Forgets the Location of something- Detection



User keeps looking inside a compartment for a long time



Implementation Details

```
import RPi.GPIO as GPIO

#Set Broadcom mode so we can address GPIO pins by number
GPIO.setmode(GPIO.BCM)

#This is the GPIO pin number we have one of the door sensor
#wires attached to, the other should be attached to a ground
DOOR_SENSOR_PIN_ONE = 18
DOOR_SENSOR_PIN_TWO = 12
DOOR_SENSOR_PIN_THREE = 25

#Set up the door sensor pin
GPIO.setup(DOOR_SENSOR_PIN_ONE, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(DOOR_SENSOR_PIN_TWO, GPIO.IN, pull_up_down = GPIO.PUD_UP)
GPIO.setup(DOOR_SENSOR_PIN_THREE, GPIO.IN, pull_up_down = GPIO.PUD_UP)

while True:
    oldIsOpenOne = isOpenOne
    isOpenOne = GPIO.input(DOOR_SENSOR_PIN_ONE)

    oldIsOpenTwo = isOpenTwo
    isOpenTwo = GPIO.input(DOOR_SENSOR_PIN_TWO)

    oldIsOpenThree = isOpenThree
    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)

Question 1

a) Elena
b) Bethania
c) Inés
d) Juana

Question 2

a) Ester
b) Meritxell
c) Amalia
d) Agustina

Question 3

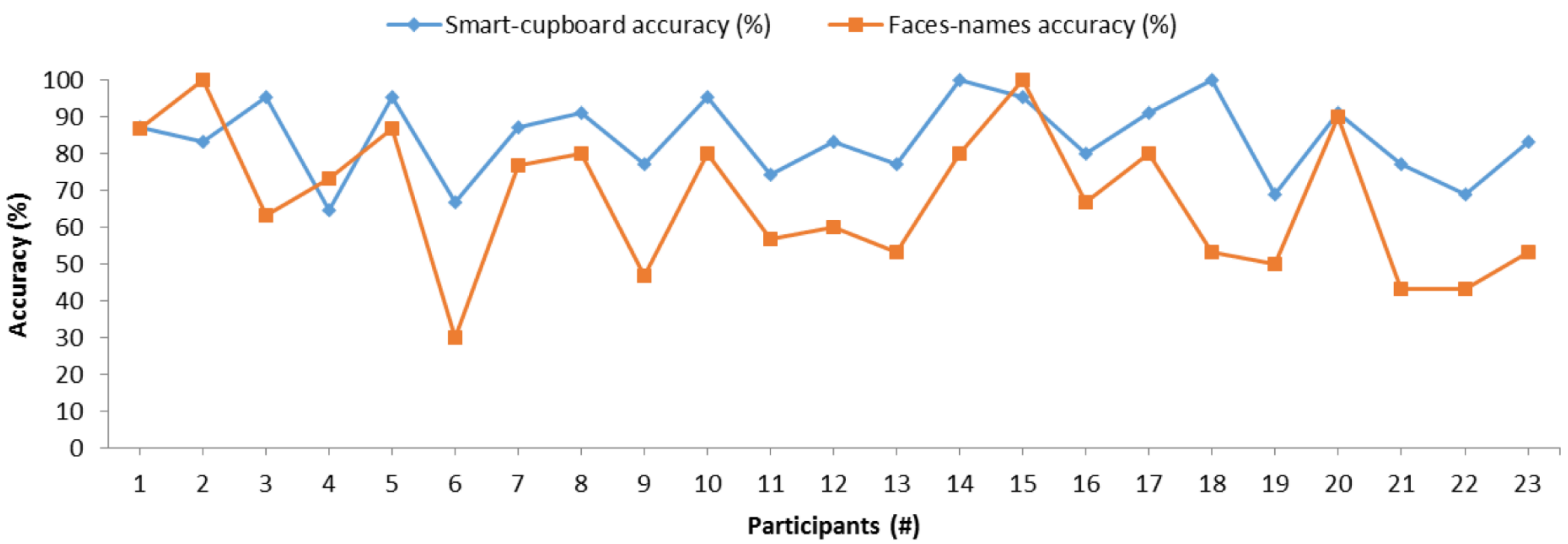
a) Hugo
b) Cirino
c) Xesús
d) Vidal

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



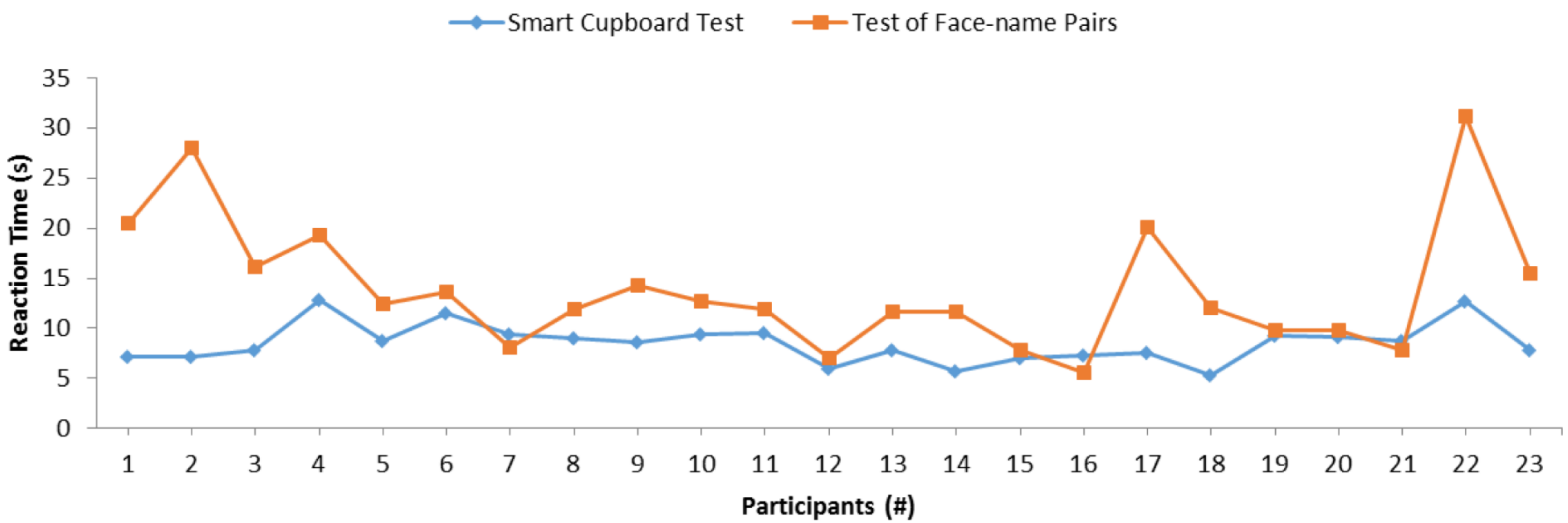
Comparison of Accuracies of Smart cupboard and Face-names

		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

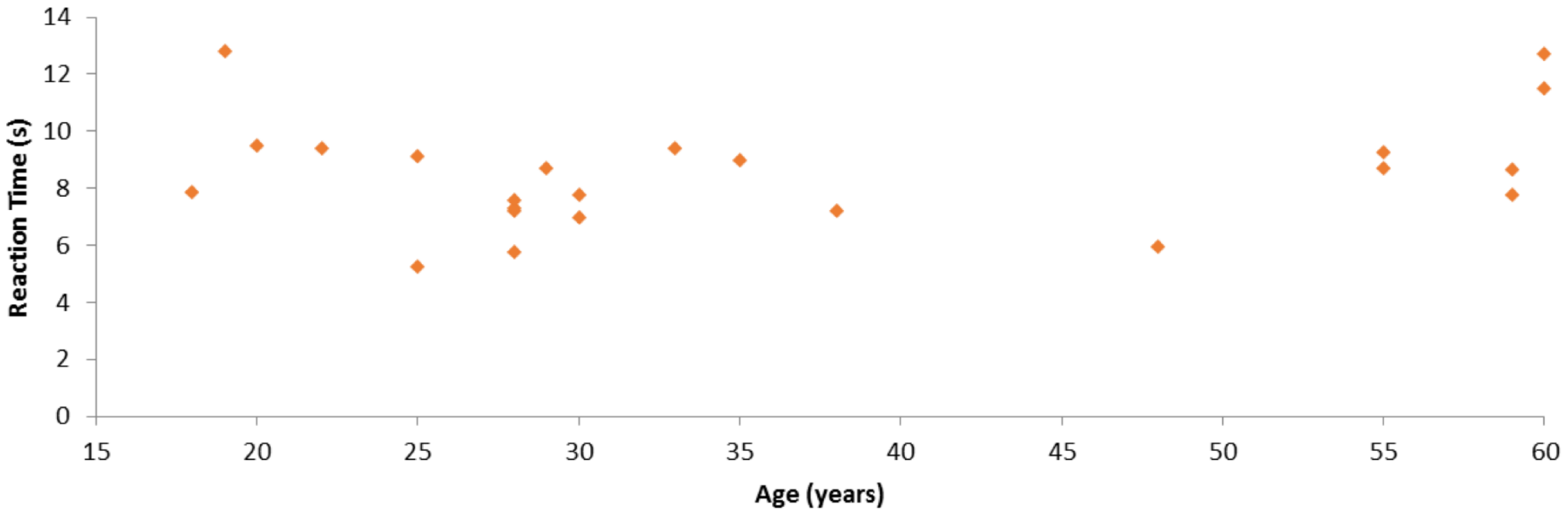


Comparison of Reaction Times

		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

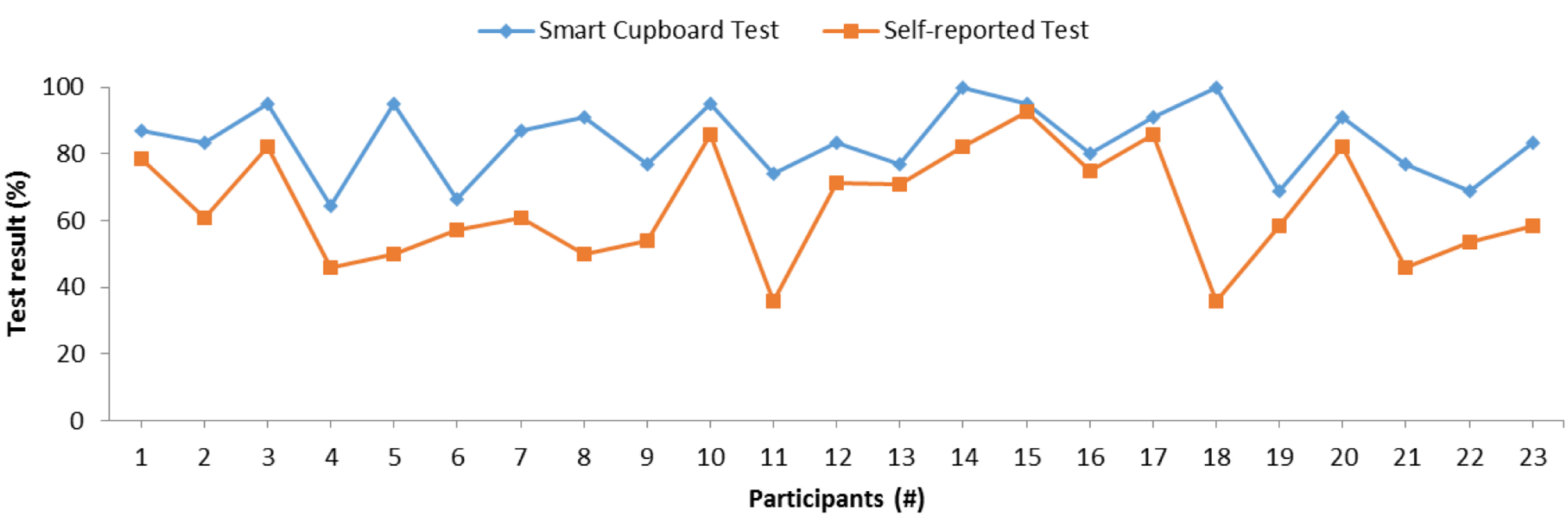


Reaction time in smart cupboard and Age

		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

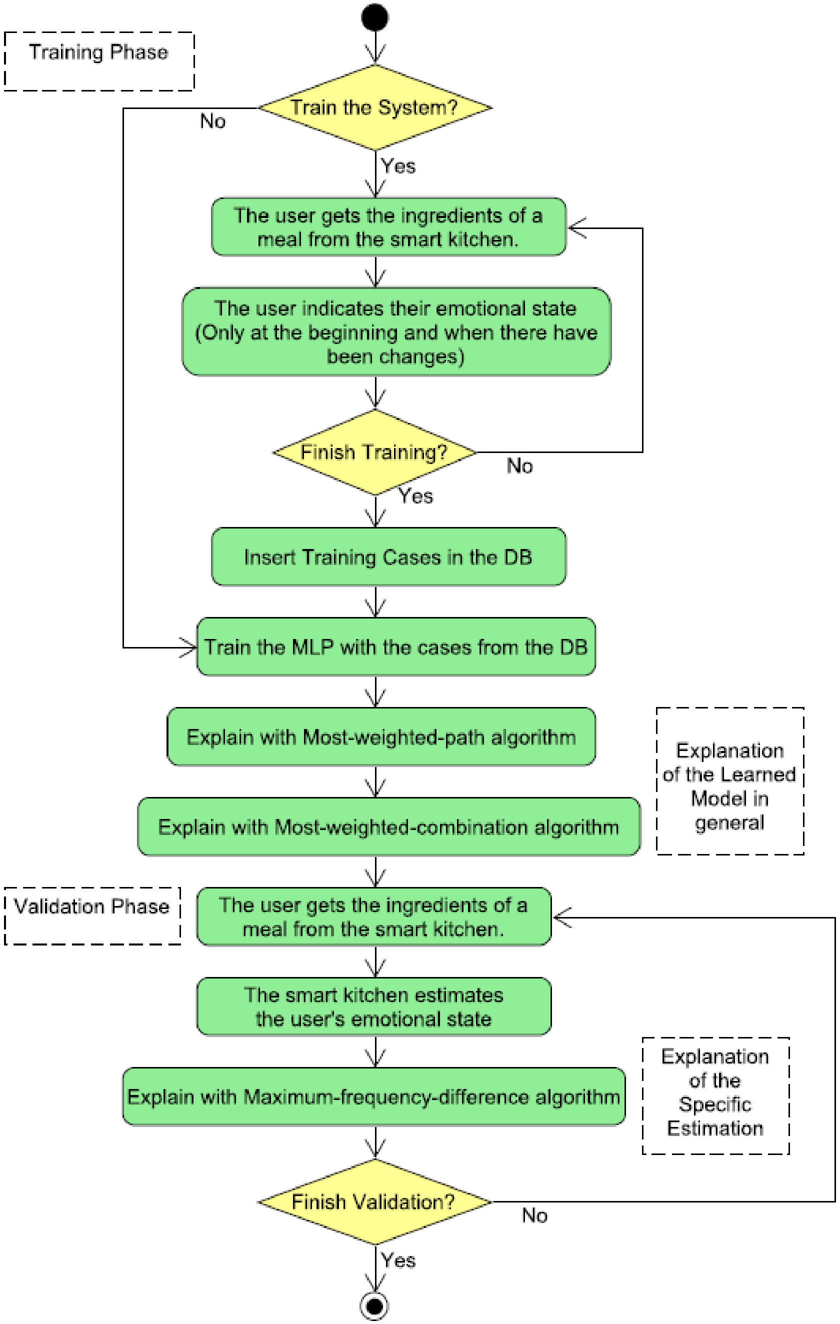
		Smart Cupboard	Accuracy Self-Reported test
Smart Cupboard	Pearson Correlation	1	.443*
	Sig. (2-tailed)		.034
	N	23	23
Accuracy Self-Reported test	Pearson Correlation	.443*	1
	Sig. (2-tailed)	.034	
	N	23	23

*. 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

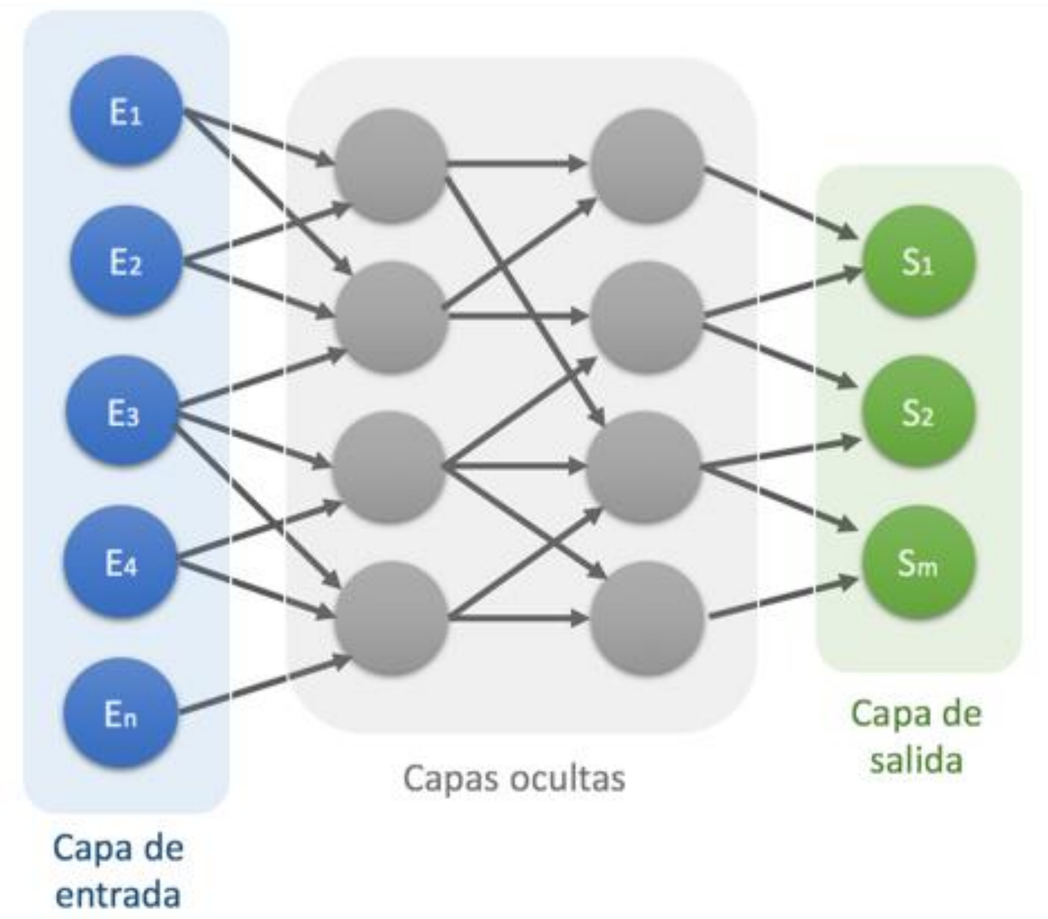


Approach for experimenting HAI algorithms in simulated smart kitchens

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

```
function explainMostWeightedPath(mlp, names)
    current ← mlp.outputNeuron
    while mlp.isNeuron(current) do
        dendrite ← mostWeightedDendrite(neuron)
        current ← mlp.connectedTo(dendrite)
    input ← current
    inputName ← names.inputs[input]
    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+'.'
    return explanation
```

Esquema de Red Neuronal MLP



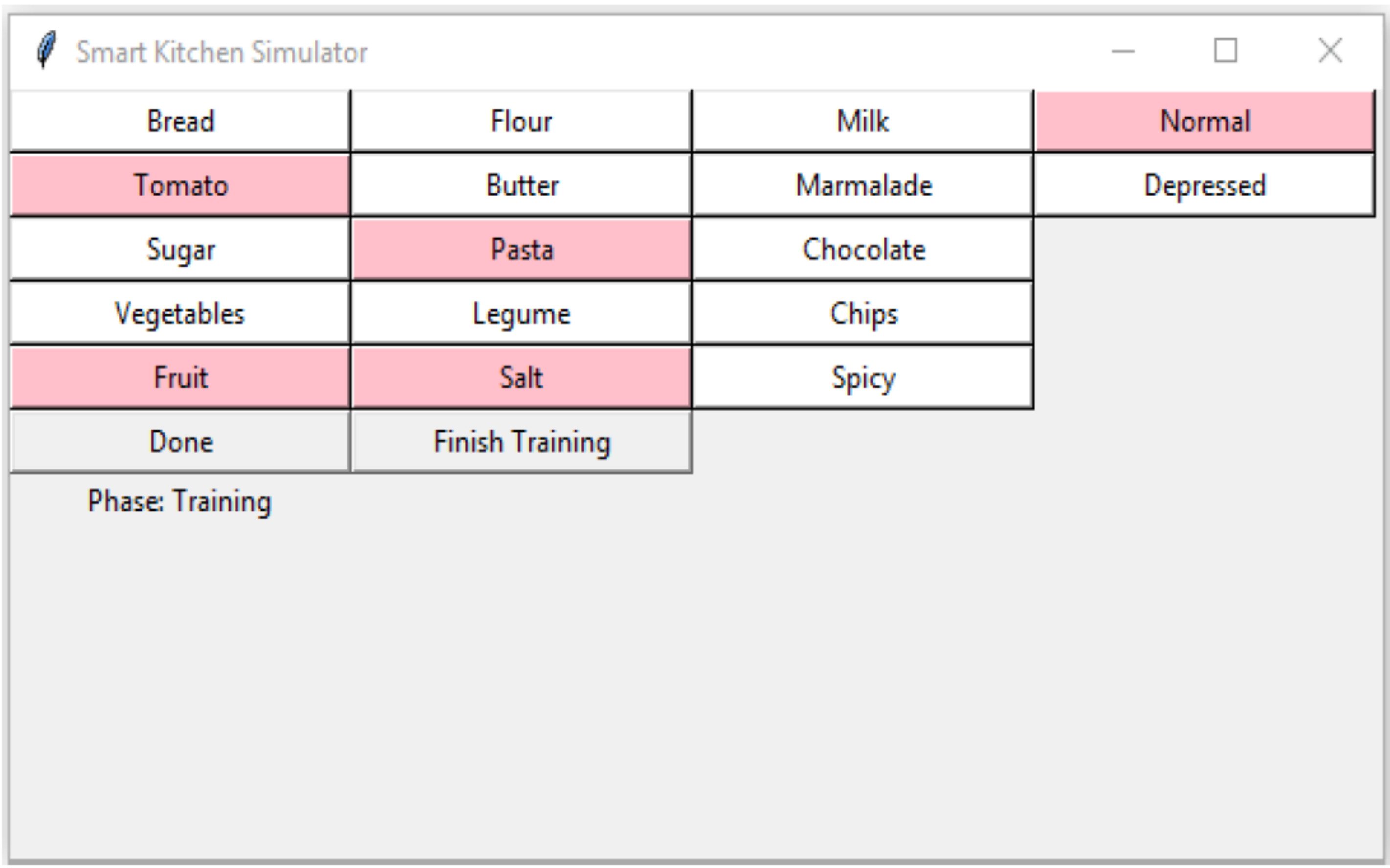
Algorithm 2 Most-Weighted-Combination Explanation: It Provides a HAI Explanation Based on the Most Relevant Combination of Two Inputs Based the Most Weighted Dendrites of the First and Second Neuron Layers

```
function explainMostWeightedCombination(mlp, names)
    layer ← 1            ▷ Second layer, as count starts on 0
    numInputs ← 2
    dendrites ← mlp.getDentritesLayer(layer)
    quicksortByWeightDescendentOrder(dendrites)
    found ← false
    i ← 0
    while i<dendrites.length and not found do
        dendrite ← dendrites[i]
        inputNeuron ← mlp.connectedTo(dendrite)
        inputDendrites ← mlp.mostWeightedDendrites(
                        neuron, numInputs)
        found ← true
        for j ∈ [0, numInputs) do
            if inputDendrites[j].weight>threshold then
                combination[j] ← mlp.connectedTo(
                        inputDendrites[j])
            else
                found ← false
        i ← i + 1
    if found then
        explanation ← ‘In the learned model for the ‘+names.IoTsystem+’, the most relevant input combination for estimating whether you are ‘+names.highestOutputValue+’ is that ‘+names.userAction+’ ‘+ names.inputs[combination[0]]+’ and ‘+names.inputs[combination[1]]+’.’
    else
        explanation ← ‘No combination of two inputs is especially relevant.’
    return explanation
```

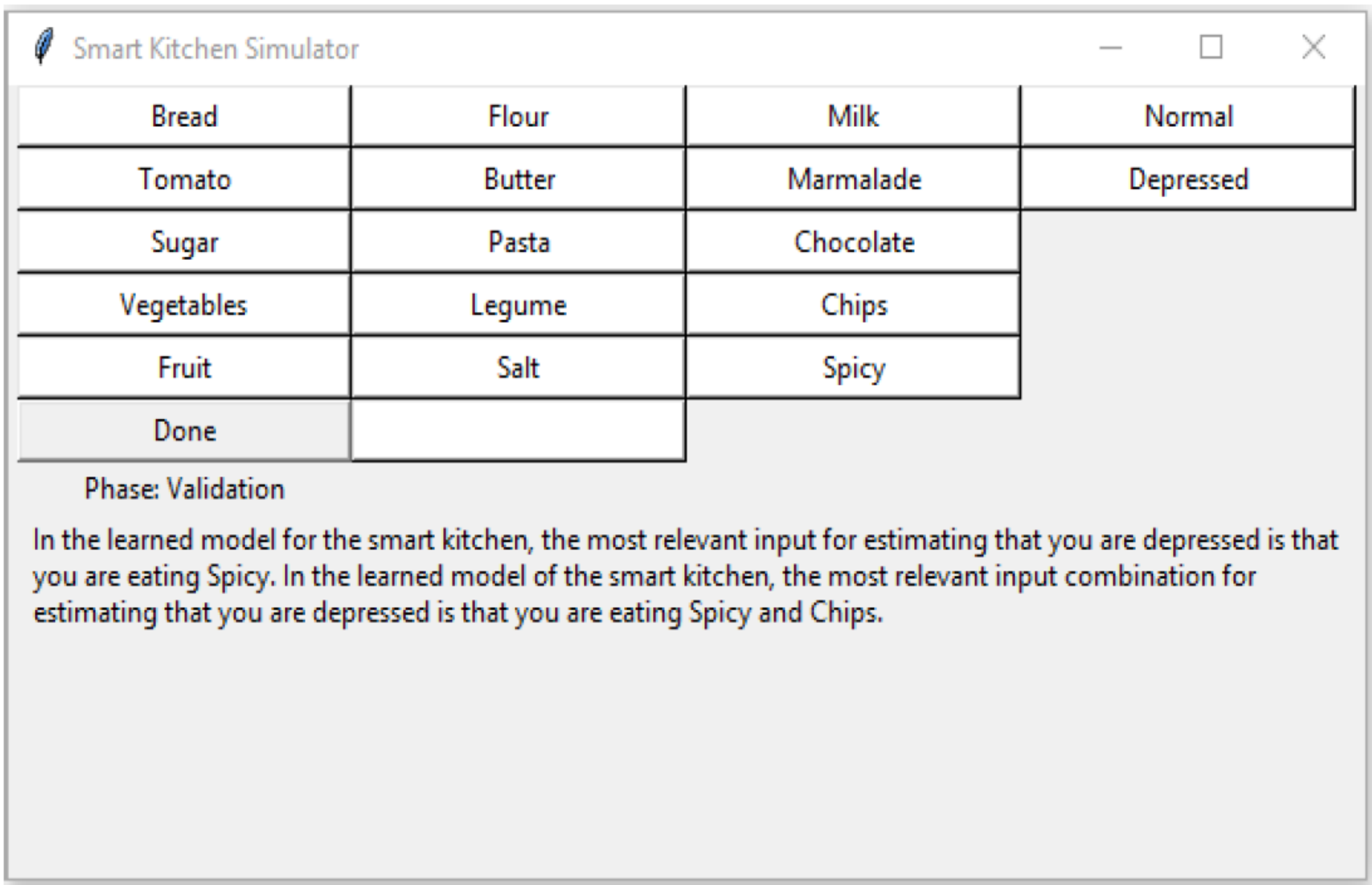
Algorithm 5 Maximum-Frequency-Difference Explanation: It Provides a HAI Explanation Based on the Most Discriminative Input, Measured as the One With the Highest Difference of Frequency Percentage for the Given Prediction

```
function explainMaxFreqDiff(caseInputs, prediction, names)
    maxDiff ← minIntValue
    maxInputName ← ‘’
    for i ∈ [0,names.inputs.length) do
        if caseInputs[i] then
            inputName ← names.inputs[i]
            diff ← DiffPercen(inputName,prediction)
            if diff>maxDiff then
                maxInputName ← inputName
                maxDiff ← diff
    explanation ← ‘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+’.’
    return explanation
```

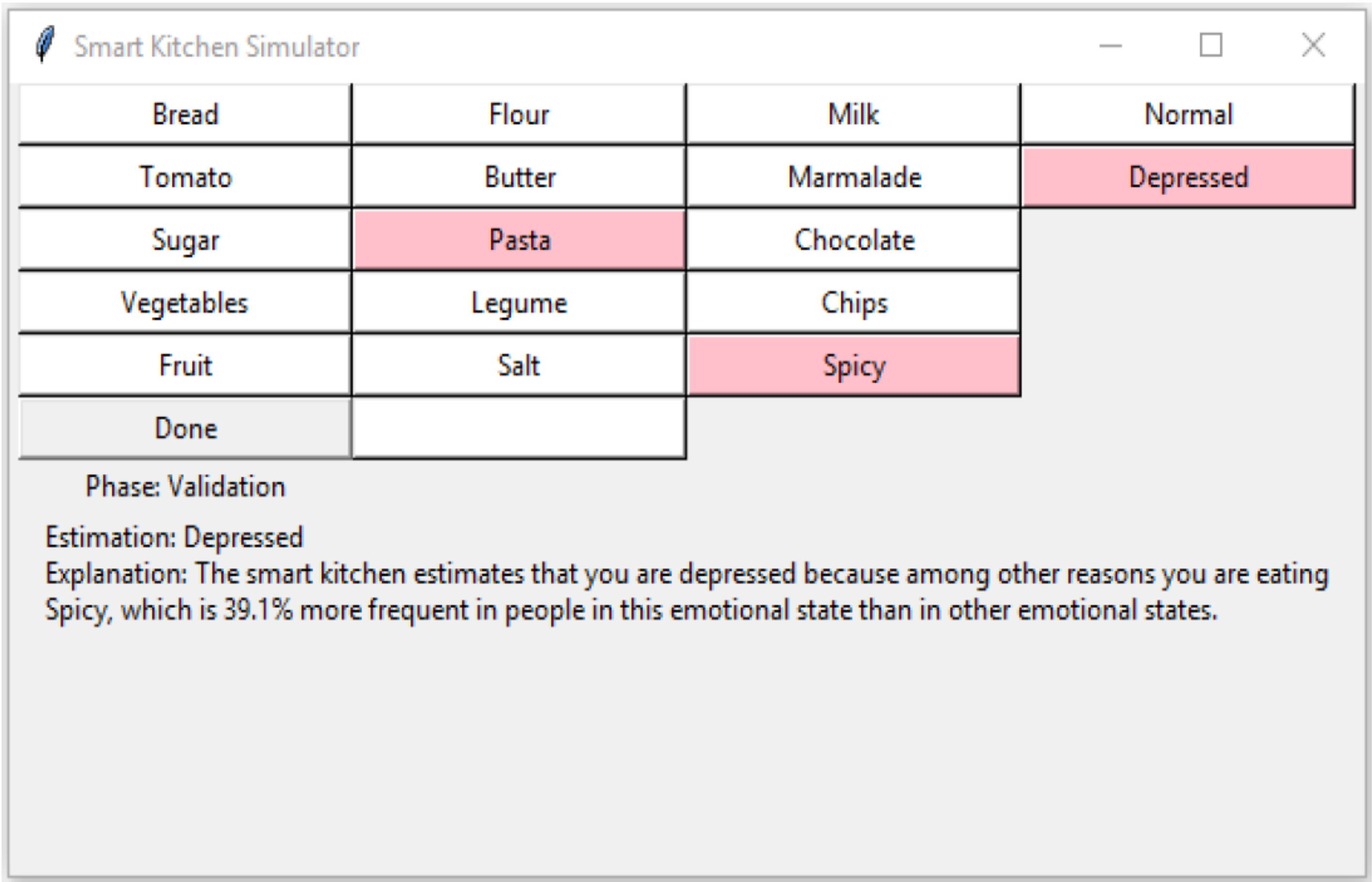
Smart Kitchen Simulator (training phase)



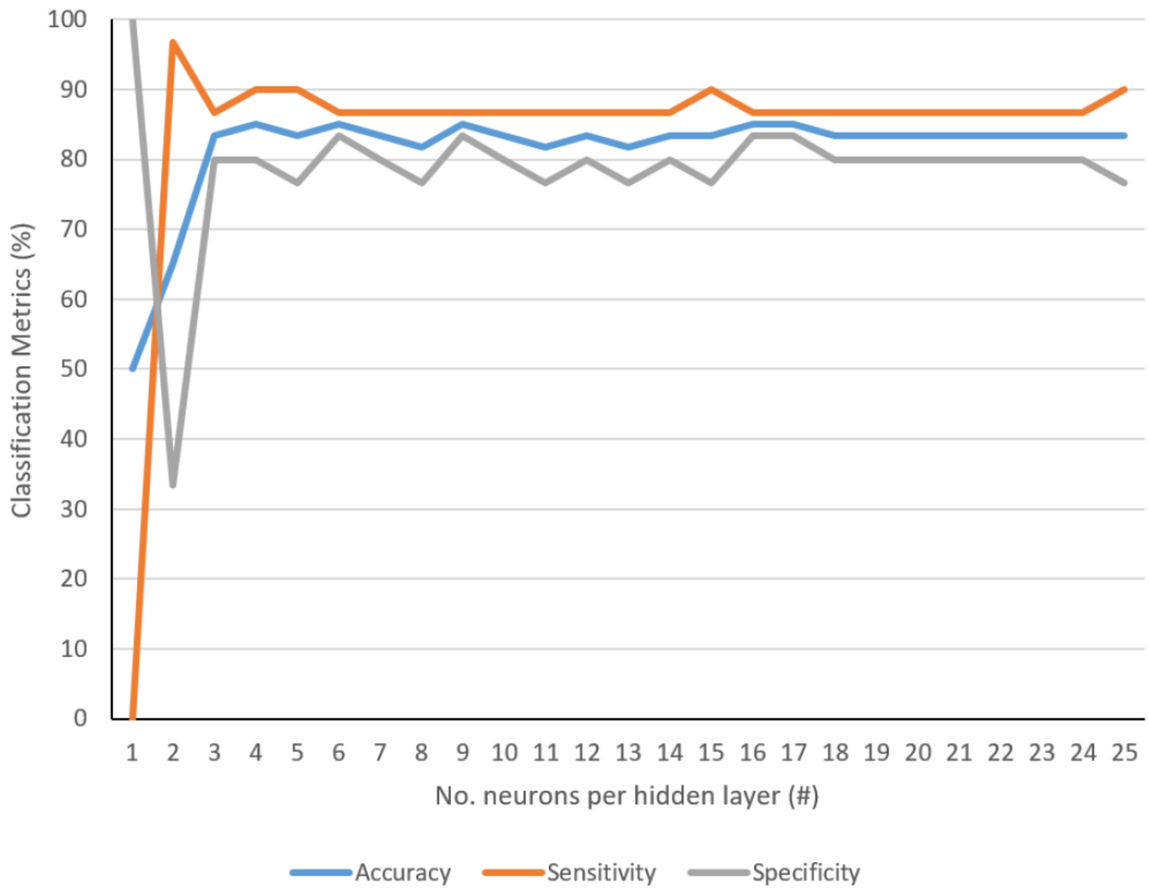
Auto-generated explanation for the learned model

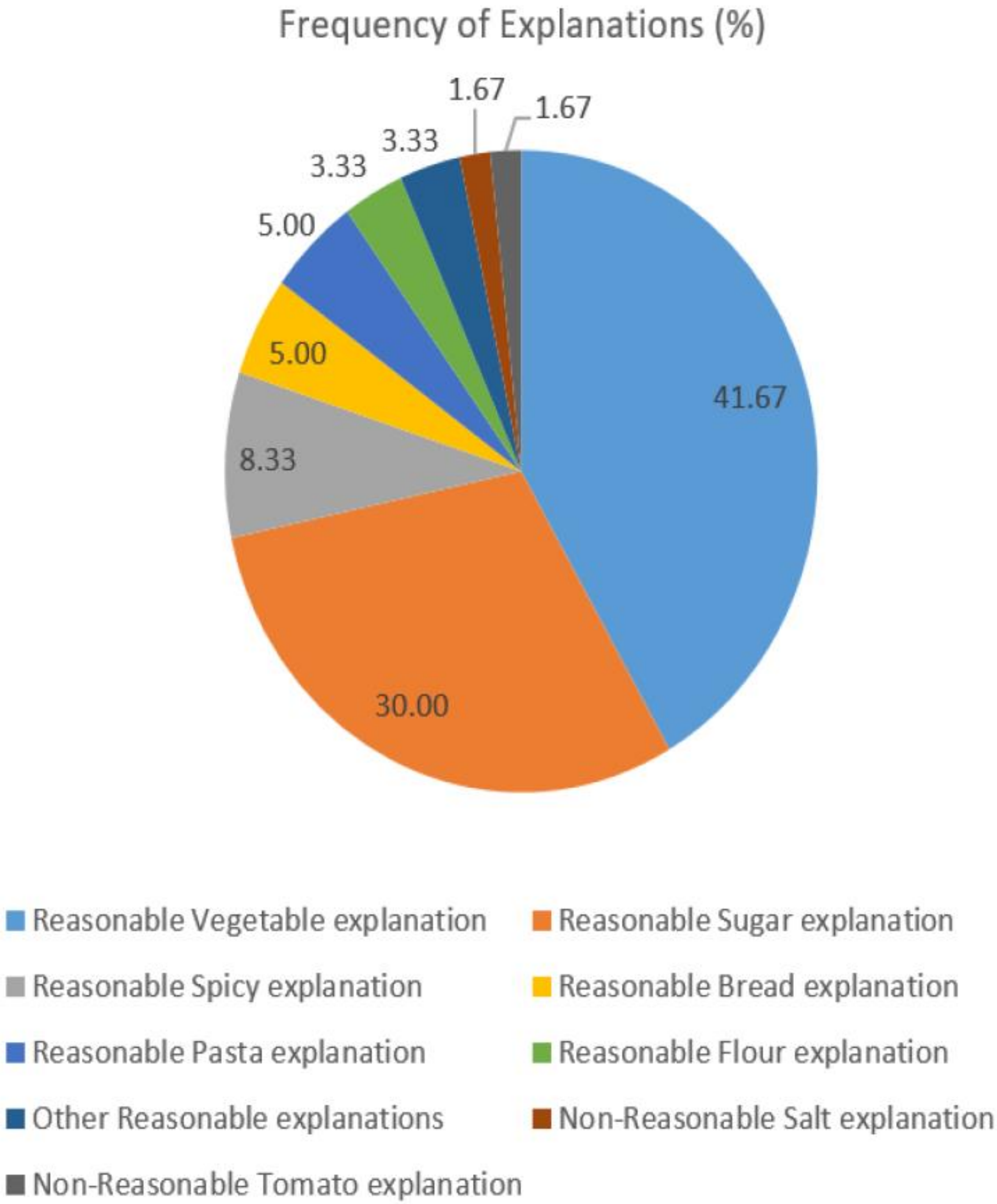


Auto-generated explanation for one meal



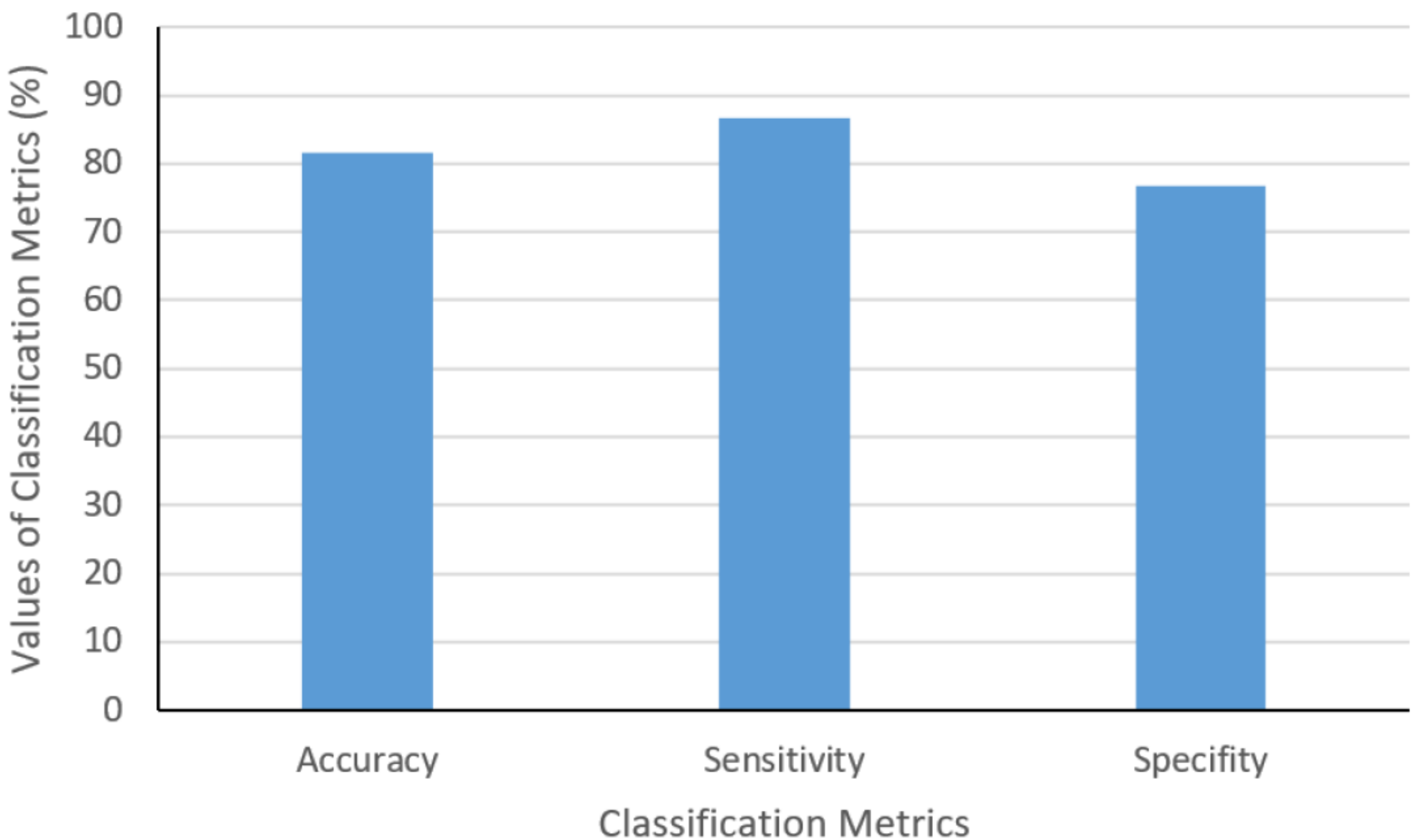
Adjusting Neural Network parameters





Results about HAI auto-generated explanations

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