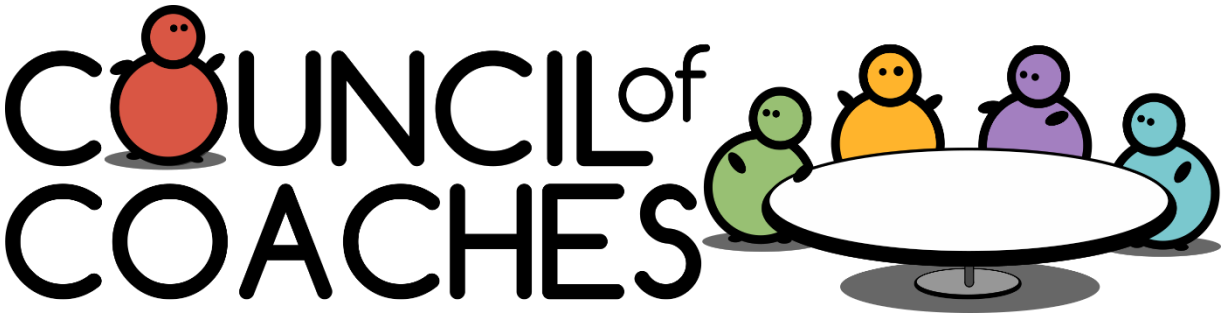


D4.2: Methods for inferring short-term behaviour from multimodal sensor data

Dissemination level:	Public
Document type:	Report
Version:	1.0.1
Date:	August 31, 2018 (original)
	March 5, 2019 (this version)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement #769553. This result only reflects the author's view and the EU is not responsible for any use that may be made of the information it contains.

Document Details

Project Number	769553
Project title	Council of Coaches
Title of deliverable	Methods for inferring short-term behaviour from multimodal sensor data
Due date of deliverable	August 30, 2018
Work package	WP4
Author(s)	Oresti Banos (CMC), Kostas Konsolakis (CMC), Harm op den Akker (RRD), Catherine Pelachaud (SU), Reshmashree Bangalore (SU)
Reviewer(s)	Tessa Beinema (RRD)
Approved by	Coordinator
Dissemination level	Public
Document type	Report
Total number of pages	40

Partners

- University of Twente – Centre for Monitoring and Coaching (CMC)
- Roessingh Research and Development (RRD)
- Danish Board of Technology Foundation (DBT)
- Sorbonne University (SU)
- University of Dundee (UDun)
- Universitat Politècnica de València, Grupa SABIEN (UPV)
- Innovation Sprint (iSPRINT)

Abstract

This deliverable (D4.2) represents the contribution of the Work Package 4 (WP4) to the inference of short-term behaviours from physical and virtual sensor data. Short-term behaviours are investigated, as part of the task T4.1, in order to define the most optimal real-time coaching strategies that can be provided through the Council of Coaches project. This document investigates and explains the methods used for the automatic processing of on-body and off-body sensor data in order to measure and model the user's behaviour. Hence, different types of sensor data are thoroughly investigated in order to develop short-term behaviour models. Finally, the evaluation of the developed methods and the primary results are presented.

Corrections

v1.0.1 Correctly applied EU logo on header page.

Table of Contents

List of figures

List of tables

Symbols, abbreviations and acronyms

ACC	Accelerometer
AU	Action Unit
CMC	Centre for Monitoring and Coaching
COUCH	Council of Coaches
D	Deliverable
EC	European Commission
FFT	Fast Fourier Transform
GPS	Global Positioning System
GSR	Galvanic Skin Response
HBAF	Holistic Behaviour Analysis Framework
HCI	Human-Computer Interaction
KNN	k-Nearest Neighbour
M	Month
Max	Maximum
Min	Minimum
MS	Milestone
PPG	Photoplethysmography
RF	Random Forest
RMS	Root Mean Square
RMSE	Root Mean Square Error
RRD	Roessingh Research and Development
STD	Standard Deviation
SVM	Support Vector Machine
WP	Work Package
UDI	Unique Device Identifier
UUI	Unique User Identifier
UPMC	Université Pierre et Marie Curie, Paris 6
UPV	Universitat Politècnica de València
UT	University of Twente

1 Introduction

One of the tasks of the Council of Coaches project focuses on combining smart multimodal sensing technologies to seamlessly and opportunistically detect user's behaviour, including physical, cognitive, mental and social aspects. After presenting the initial concept of the Holistic Behaviour Analysis Framework (HBAF) in D4.1 (Oresti Banos, 2018), this deliverable deepens more on the inference of short-term behaviours from sensor data. In particular, the current D4.2 document aims to investigate and explain the methods used for the automatic processing of on-body and off-body sensor data in order to measure and model the user's behaviour.

The methods used for detecting short-term behaviours (a.k.a. primitives or momentary behaviours), including the types of sensor data and the techniques for behaviour analysis are thoroughly investigated and presented in Section 3. The evaluation of these methods, including the techniques for the data collection and analysis, is elaborated in Section 4. Finally, Section 5 defines the main outcomes of this deliverable with respect to the coaching strategies of the Council of Coaches.

2 Objectives

The main objective of this deliverable (D4.2) is to describe the methods developed for the inference of short-term behaviours based on multimodal sensor data. Accordingly, this document aims to investigate and elaborate on the sensor data types and processing techniques required to detect a relevant set of short-term behaviours, which have been found necessary to support different components of Council of Coaches (Gerwin Huizing, 2018).

3 Short-term Behaviour Analysis Methods

3.1 Short-term Behaviours

Short-term behaviours or primitives refer to physical, emotional, social or cognitive behaviours that take place under a limited period of time. This period of time can range from minutes to hours, depending on the behaviour type and its use in the proposed coaching strategies.

In our attempt to provide data that can be used to help to realise some coaching strategies, we have considered the following two scenarios. For the first scenario, a user with diabetes type 2 requests to monitor physical activity every minute in order to effectively contextualise the changes in glucose level and receive the necessary coaching services for his diet. The same user gives access to the Council of Coaches to monitor his social, emotional and cognitive behaviour every three hours. For the second scenario, an elderly user with age related impairments wants to receive coaching services for the cognitive and social behaviour every 6 hours. From these two scenarios, it is clear that the definition of short-term behaviours relies on the nature of the behaviour but also on the task of monitoring. For instance, walking or talking with another person for a few seconds is considered a short-term physical and social behaviour, respectively. On the contrary, a sequence of short-term behaviours over a longer period of time describes the so-called behavioural patterns (routines) of the user, which will be thoroughly investigated in task T4.2.

Short-term physical behaviour refers here to the detection of physical activities, such as sitting, standing, walking and cycling, and the detection of steps. Primitives of social behaviour are related to the socialisation level of a user (user's interaction with other individuals vs. user's isolation), emotional behaviour refers to the monitoring of user's emotions and mood (e.g. happy, sad, bored, stressed), while cognitive behaviour concerns the user's attentiveness during a conversation (with a virtual coach). The aforementioned types of short-term behaviour were carefully selected based on the project demands and based on the available sensor sources for providing the necessary coaching strategies. A thorough investigation of these sources was presented in deliverable D4.1 (Oresti Banos, 2018).

Multimodal behaviours are multi-functions. They can signal emotions, prominence, social attitude, engagement, cognitive load to name a few of these communicative functions. To categorize these functions, several representations have been proposed by scholars.

Affect can be described in terms of discrete (basic) emotional categories that include happiness, sadness, fear, anger, disgust, and surprise (Zeng & Pantic, 2009). However, a discrete list of emotions fails to describe the range of emotions that occur during interactions. An alternative way to describe this is using the dimensional description (Ekman., 1997), see Figure 1 below. The most widely used representation of affective states is in terms of dimensions of evaluation (valence) and activation (arousal) and some studies use an additional dimension, dominance (Kroschel., 2005). Evaluation (valence) measures how humans feel, from positive to negative and activation (arousal) measures whether humans are more or less likely to take an action under the emotional state, from active to passive. For this project, we will be measuring the arousal and the valence of the users to measure affect.

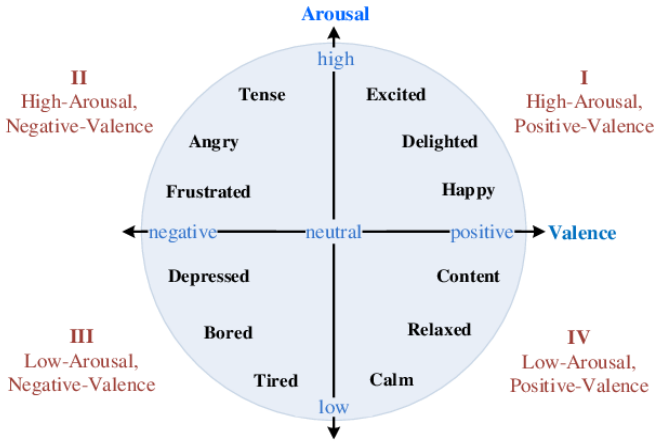


Figure 1: Two-dimensional valence-arousal space (Yu, 2016).

During an interaction, participants display various multimodal behaviours that are temporally aligned. These behaviours are linked to how the interaction between the participants evolves. An important aspect of human-agent interaction is engagement which ensures the interaction to move forwards and can be categorised as a cognitive behaviour. A detailed summary of engagement definitions in human-agent interaction is provided in (Glas, 2015). Engagement is defined as: *"the value that a participant in an interaction attributes to the goal of being together with the other participant(s) and of continuing the interaction"* (Poggi, 2007). It is also defined as the process by which participants involved in an interaction start, maintain and terminate an interaction (Sidner C. L., 2005). Engagement is not measured from single cues, but rather from several cues that arise over a certain time window (Peters, 2005). Engagement can be defined by high-level behaviour like, *synchrony* – which is temporal coordination during social interactions; *mimicry* – which is automatic tendency to imitate others; *feedback* – which can indicate whether the communication is successful or not. Similarly, engagement can also be defined by low-level behaviour like *eye gaze* - providing feedback and showing interest; *head movements* - nods (in agreement, disagreement, in between); *gestures* - to greet, to take turns; *postures* - body orientation, lean; *facial expressions*.

The behaviour analysis model will be used as a means to feed coaching actions, steer dialogues between the user and the Council of Coaches but also steer a non-verbal interaction between the user and the system. For instance, physical behaviour primitives will be used in order to monitor user's health and well-being, but also to monitor chronic diseases (e.g., stroke, arthritis, Parkinson, diabetes) over time. Short-term emotional behaviours will be analysed in order to monitor user's emotional states and mood, but also to track stress and anxiety over time, and provide the relevant coaching strategies. Social primitives will be used to help defining the social life of a user and investigate in a later phase (T4.2) the consequences of being socially isolated for a longer period of time. Short-term cognitive behaviours will monitor the user's level of alertness, attendance and fatigue, in order to steer the dialogues and enhance the system decisions for coaching strategies.

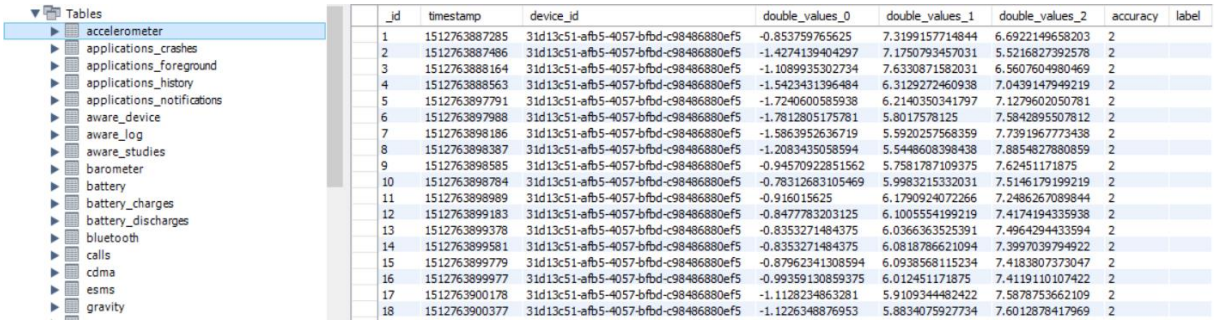
3.2 Sensor Data

The inference of short-term behaviour is based on acquired data from on-body and off-body sensors. On-body sensors refer to acquired data through smartphones in order to analyse short-term physical and social behaviour. In particular, a model for detecting steps count, predicting the performed activities (walking, tilting, sitting, cycling, and taking the bus), and detecting the level of user's social interaction and isolation will be developed.

Smartphone data consist of accelerometer and GPS signals for tracking physical activity, while Bluetooth, ambient noise, location and phone usage metrics (e.g., number of incoming/outcoming calls, number of

received/sent text messages) will be used to detect the user's level of being socially active. The aforementioned types of data are presented in the following figures (see Figure 1-7).

Accelerometer: The accelerometer sensor measures acceleration over time along three axes; axis x (double_values_0), axis y (double_values_1) and axis z (double_values_2). An example of the acquired accelerometer data is presented in Figure 2.



_id	timestamp	device_id	double_values_0	double_values_1	double_values_2	accuracy	label
1	1512763887285	31d13c51-afb5-4057-bfbd-c98486880ef5	-0.853759765625	7.3199157714844	6.6922149658203	2	
2	1512763887486	31d13c51-afb5-4057-bfbd-c98486880ef5	-1.4274139404297	7.1750793457031	5.5216827392578	2	
3	1512763888164	31d13c51-afb5-4057-bfbd-c98486880ef5	-1.1089935302734	7.6330871582031	6.5607604980469	2	
4	1512763888563	31d13c51-afb5-4057-bfbd-c98486880ef5	-1.5423431396484	6.3129272460938	7.0439147949219	2	
5	1512763897791	31d13c51-afb5-4057-bfbd-c98486880ef5	-1.7240600585938	6.2140350341797	7.1279602050781	2	
6	1512763897988	31d13c51-afb5-4057-bfbd-c98486880ef5	-1.7812805175781	5.8017578125	7.5842895507812	2	
7	1512763898186	31d13c51-afb5-4057-bfbd-c98486880ef5	-1.5863952636719	5.5920257568359	7.7391967773438	2	
8	1512763898387	31d13c51-afb5-4057-bfbd-c98486880ef5	-1.2083435058594	5.5448608398438	7.8854827880859	2	
9	1512763898585	31d13c51-afb5-4057-bfbd-c98486880ef5	-0.94570922851562	5.7581787109375	7.62451171875	2	
10	1512763898784	31d13c51-afb5-4057-bfbd-c98486880ef5	-0.78312683105469	5.9983215332031	7.5146179199219	2	
11	1512763898989	31d13c51-afb5-4057-bfbd-c98486880ef5	-0.916015625	6.1790924072266	7.2486267089844	2	
12	1512763899183	31d13c51-afb5-4057-bfbd-c98486880ef5	-0.8477783203125	6.1005554199219	7.4174194335938	2	
13	1512763899378	31d13c51-afb5-4057-bfbd-c98486880ef5	-0.8353271484375	6.0366363525391	7.4964294433594	2	
14	1512763899581	31d13c51-afb5-4057-bfbd-c98486880ef5	-0.8353271484375	6.0818786621094	7.3997039794922	2	
15	1512763899779	31d13c51-afb5-4057-bfbd-c98486880ef5	-0.87962341308594	6.0938568115234	7.4183807373047	2	
16	1512763899977	31d13c51-afb5-4057-bfbd-c98486880ef5	-0.99359130859375	6.012451171875	7.4119110107422	2	
17	1512763900178	31d13c51-afb5-4057-bfbd-c98486880ef5	-1.1128234863281	5.9109344482422	7.5878753662109	2	
18	1512763900377	31d13c51-afb5-4057-bfbd-c98486880ef5	-1.1226348876953	5.8834075927734	7.6012878417969	2	

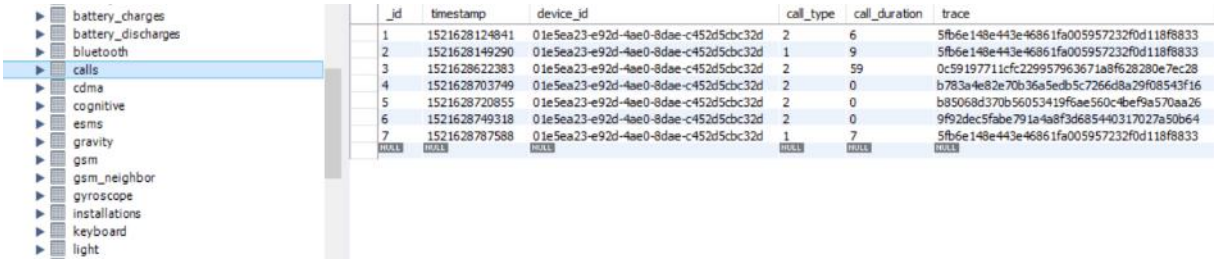
Figure 2: Example of acquired accelerometer data stored in the database.

Bluetooth: The Bluetooth sensor detects surrounding Bluetooth-enabled and visible devices with respective RSSI values at specified intervals (default is 60 seconds). The signal gets stronger when the RSSI values are closer to zero. RSSI is expressed in decibels relative to a milliwatt (dBm) from zero to -120 dBm. An example of the acquired Bluetooth data is presented in Figure 3.

_id	timestamp	device_id	bt_address	bt_name	bt_rssi	label
1	1512764839427	31d13c51-afb5-4057-bfbd-c98486880ef5	40:F0:2F:95:CF:F5	TC-CZC4074B8D	-88	1512764832172
2	1512765249139	31d13c51-afb5-4057-bfbd-c98486880ef5	00:26:7E:31:82:25		-84	1512765239225
3	1512765313360	31d13c51-afb5-4057-bfbd-c98486880ef5	10:C6:FC:C6:90:6A		-86	1512765302949
4	1512765313931	31d13c51-afb5-4057-bfbd-c98486880ef5	B4:8B:19:DA:02:1E		-98	1512765302949
5	1512765702197	31d13c51-afb5-4057-bfbd-c98486880ef5	77:95:AE:47:24:3C		-87	1512765698338
6	1512765834595	31d13c51-afb5-4057-bfbd-c98486880ef5	68:27:37:0E:8C:47		-89	1512765832439
7	1512766015229	31d13c51-afb5-4057-bfbd-c98486880ef5	B8:BB:AF:C2:B3:45	[TV] Samsung 5 Series (32)	-85	1512766013404
8	1512766016607	31d13c51-afb5-4057-bfbd-c98486880ef5	14:6F:0E:7A:F8:F8		-92	1512766013404
9	1512766017314	31d13c51-afb5-4057-bfbd-c98486880ef5	C8:69:CD:07:DA:C7		-87	1512766013404
10	1512766108606	31d13c51-afb5-4057-bfbd-c98486880ef5	5C:37:C5:0E:4F:16		-91	1512766099438
11	1512766109750	31d13c51-afb5-4057-bfbd-c98486880ef5	9C:8D:7C:2C:D6:09		-90	1512766099438
12	1512766111484	31d13c51-afb5-4057-bfbd-c98486880ef5	9C:8D:7C:2C:D6:09	VW BT 5970	-86	1512766099438
13	1512766168466	31d13c51-afb5-4057-bfbd-c98486880ef5	AC:BC:32:6C:B3:01		-90	1512766159588
14	1512766267372	31d13c51-afb5-4057-bfbd-c98486880ef5	08:EF:3B:94:6B:6C	LG SH2(6C)	-67	1512766265113
15	1512766452448	31d13c51-afb5-4057-bfbd-c98486880ef5	00:26:5F:A7:FD:86	S5230	-88	1512766446404
16	1512766682071	31d13c51-afb5-4057-bfbd-c98486880ef5	60:03:08:C4:33:B9		-95	1512766676393
17	1512766684619	31d13c51-afb5-4057-bfbd-c98486880ef5	94:B2:CC:03:9A:E2	DEH-4800BT	-93	1512766676393
18	1512766685581	31d13c51-afb5-4057-bfbd-c98486880ef5	1C:1A:C0:A6:AA:BC		-91	1512766676393

Figure 3: Example of acquired Bluetooth data stored in the database.

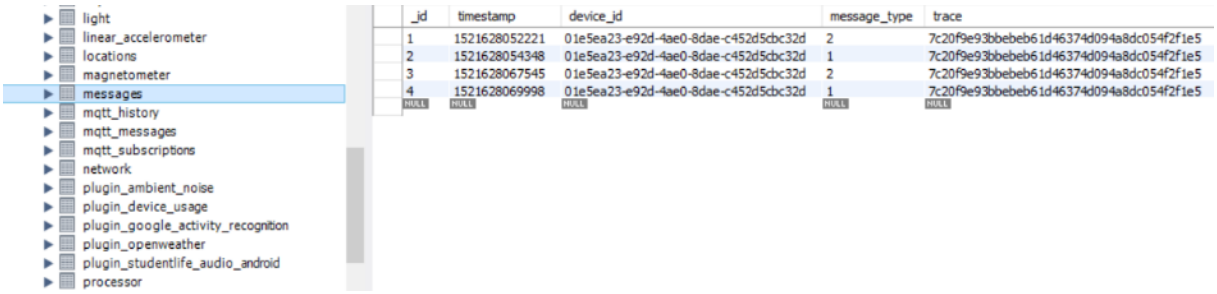
Calls: The calls sensor logs incoming and outcoming call events, performed by or received by the user. This sensor does not record personal information, such as phone numbers or contact information, but uses a unique ID based on SHA-1 encryption. An example of the acquired calls data is presented in Figure 4.



_id	timestamp	device_id	call_type	call_duration	trace
1	1521628124841	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	2	6	5fb6e148e443e46861fa005957232f0d118f8833
2	1521628149290	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	1	9	5fb6e148e443e46861fa005957232f0d118f8833
3	1521628622383	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	2	59	0c59197711cfc229957963671a8f628280e7ec28
4	1521628703749	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	2	0	b783a4e82e70b36a5edb5c7266d8a29f08543f16
5	1521628720855	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	2	0	b85068d370b56053419f6ae560c4bef9a570aa26
6	1521628749318	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	2	0	9f92dec5fabe791a4a8f3d6854403170 27a50b64
7	1521628787588	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	1	7	5fb6e148e443e46861fa005957232f0d118f8833
NULL	NULL	NULL	NULL	NULL	NULL

Figure 4: Example of acquired calls data stored in the database.

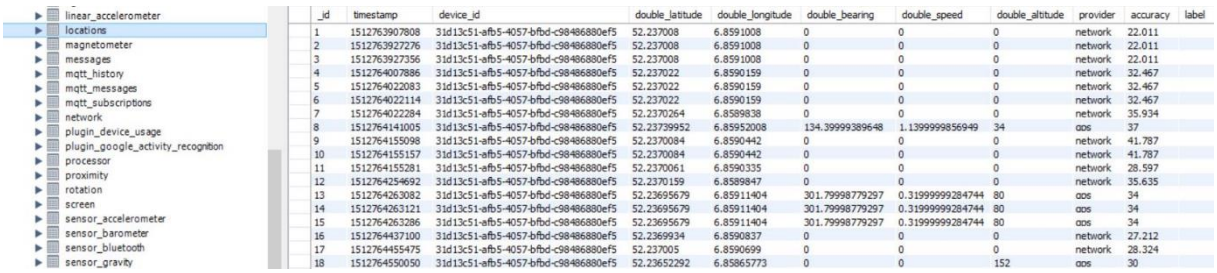
Messages: The messages sensor logs received and sent SMS texts, performed by or received by the user. This sensor does not record personal information, such as phone numbers or contact information, but uses a unique ID based on SHA-1 encryption. An example of the acquired messages data is presented in Figure 5.



_id	timestamp	device_id	message_type	trace
1	1521628052221	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	2	7c20f9e93bbebeb61d46374d094a8dc054f2f1e5
2	1521628054348	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	1	7c20f9e93bbebeb61d46374d094a8dc054f2f1e5
3	1521628067545	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	2	7c20f9e93bbebeb61d46374d094a8dc054f2f1e5
4	1521628069998	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	1	7c20f9e93bbebeb61d46374d094a8dc054f2f1e5
NULL	NULL	NULL	NULL	NULL

Figure 5: Example of acquired messages data stored in the database.

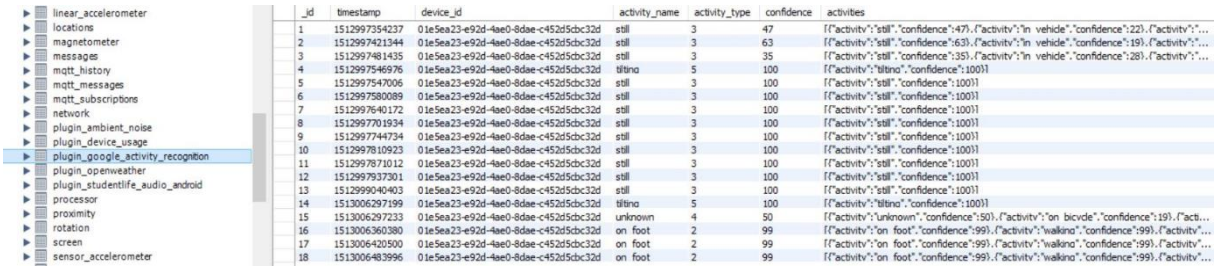
Locations: The locations sensor uses the GPS coordinates in order to provide an estimation of the user's current location, automatically. An example of the acquired locations data is presented in Figure 6.



_id	timestamp	device_id	double_latitude	double_longitude	double_bearing	double_speed	double_altitude	provider	accuracy	label
1	1512763907808	31d13c51-afb5-4057-bfbd-c98486880ef5	52.237008	6.8591008	0	0	0	network	22.011	
2	1512763927276	31d13c51-afb5-4057-bfbd-c98486880ef5	52.237008	6.8591008	0	0	0	network	22.011	
3	1512763927356	31d13c51-afb5-4057-bfbd-c98486880ef5	52.237008	6.8591008	0	0	0	network	22.011	
4	1512764007886	31d13c51-afb5-4057-bfbd-c98486880ef5	52.237022	6.8590159	0	0	0	network	32.467	
5	1512764022083	31d13c51-afb5-4057-bfbd-c98486880ef5	52.237022	6.8590159	0	0	0	network	32.467	
6	1512764022114	31d13c51-afb5-4057-bfbd-c98486880ef5	52.237022	6.8590159	0	0	0	network	32.467	
7	1512764022284	31d13c51-afb5-4057-bfbd-c98486880ef5	52.2370264	6.8589838	0	0	0	network	35.934	
8	1512764141005	31d13c51-afb5-4057-bfbd-c98486880ef5	52.23739952	6.85952008	134.39999389648	1.1399999856949	34	gps	37	
9	1512764155098	31d13c51-afb5-4057-bfbd-c98486880ef5	52.2370084	6.8590442	0	0	0	network	41.787	
10	1512764155157	31d13c51-afb5-4057-bfbd-c98486880ef5	52.2370084	6.8590442	0	0	0	network	41.787	
11	1512764155281	31d13c51-afb5-4057-bfbd-c98486880ef5	52.2370061	6.8590335	0	0	0	network	28.597	
12	1512764254692	31d13c51-afb5-4057-bfbd-c98486880ef5	52.2370159	6.8589847	0	0	0	network	35.635	
13	1512764263082	31d13c51-afb5-4057-bfbd-c98486880ef5	52.23695679	6.85911404	301.79998779297	0.31999999284744	80	gps	34	
14	1512764263121	31d13c51-afb5-4057-bfbd-c98486880ef5	52.23695679	6.85911404	301.79998779297	0.31999999284744	80	gps	34	
15	1512764263286	31d13c51-afb5-4057-bfbd-c98486880ef5	52.23695679	6.85911404	301.79998779297	0.31999999284744	80	gps	34	
16	1512764437100	31d13c51-afb5-4057-bfbd-c98486880ef5	52.2369934	6.8590837	0	0	0	network	27.212	
17	1512764455475	31d13c51-afb5-4057-bfbd-c98486880ef5	52.237005	6.8590699	0	0	0	network	28.324	
18	1512764550050	31d13c51-afb5-4057-bfbd-c98486880ef5	52.23652292	6.85865773	0	0	152	gps	30	

Figure 6: Example of acquired locations data stored in the database.

Activity Recognition API: The Activity Recognition plugin uses the Google Location API, which is based on combining accelerometer and GPS data, in order to detect user's movement and the mode of transportation. The plugin can monitor per minute the following activities: walking, cycling, using a vehicle (e.g., taking the bus), tilting and being still. An example of the Google Activity Recognition data is presented in Figure 7.



_id	timestamp	device_id	activity_name	activity_type	confidence	activities
1	1512997354237	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	47	[{"activity":"still","confidence":47},{"activity":"in vehicle","confidence":22},{"activity":"...
2	1512997421344	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	63	[{"activity":"still","confidence":63},{"activity":"in vehicle","confidence":19},{"activity":"...
3	1512997481435	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	35	[{"activity":"still","confidence":35},{"activity":"in vehicle","confidence":28},{"activity":"...
4	1512997546976	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	tilting	5	100	[{"activity":"tilting","confidence":100}]
5	1512997547006	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	100	[{"activity":"still","confidence":100}]
6	1512997580089	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	100	[{"activity":"still","confidence":100}]
7	1512997640172	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	100	[{"activity":"still","confidence":100}]
8	1512997701934	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	100	[{"activity":"still","confidence":100}]
9	1512997744734	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	100	[{"activity":"still","confidence":100}]
10	1512997810923	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	100	[{"activity":"still","confidence":100}]
11	1512997871012	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	100	[{"activity":"still","confidence":100}]
12	1512997937301	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	100	[{"activity":"still","confidence":100}]
13	1512999040403	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	still	3	100	[{"activity":"still","confidence":100}]
14	1513006297199	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	tilting	5	100	[{"activity":"tilting","confidence":100}]
15	1513006297233	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	unknown	4	50	[{"activity":"unknown","confidence":50},{"activity":"on bicycle","confidence":19},{"acti...
16	1513006360380	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	on foot	2	99	[{"activity":"on foot","confidence":99},{"activity":"walking","confidence":99},{"activity"...
17	1513006420500	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	on foot	2	99	[{"activity":"on foot","confidence":99},{"activity":"walking","confidence":99},{"activity"...
18	1513006483996	01e5ea23-e92d-4ae0-8dae-c452d5cbc32d	on foot	2	99	[{"activity":"on foot","confidence":99},{"activity":"walking","confidence":99},{"activity"...

Figure 7: Example of acquired Google Activity Recognition (plugin) data stored in the database.

Ambient Noise: The ambient noise plugin measures the ambient noise based on the double_frequency (Hz), the double_decibels (dB) and the double_rms (dB). An environment is considered noisy when the perceived loudness gets louder and the RMS value gets closer to 0dB (maximum). The plugin uses audio from the microphone sensor and detects if the environment is noisy or silent for the last five minutes. An example of the ambient noise data is presented in Figure 8.

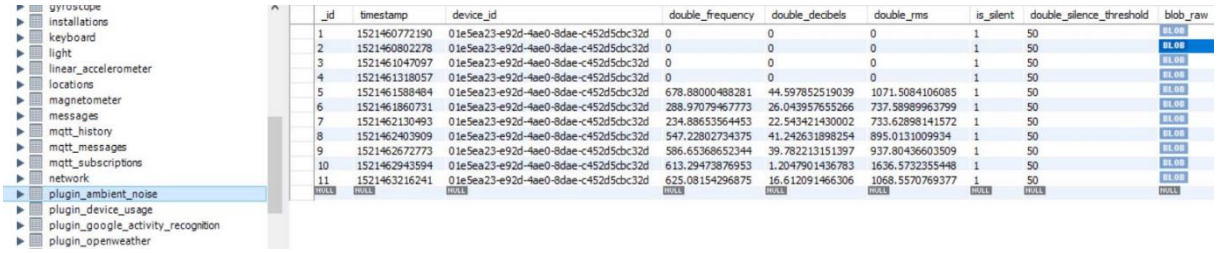


Figure 8: Example of acquired ambient noise (plugin) data stored in the database.

3.3 Techniques

In order to develop short-term behaviour models, different techniques are used in order to process raw data and extract relevant features. The data processing, as well as the whole procedure of analysing data is performed using the programming language Python (version 2.7) and particularly the python library 'scikit-learn' (scikit-learn, scikit-learn: Machine Learning in Python, 2018). Python has been highly recommended as one of the most powerful and flexible open source tools for data analysis in many classification problems (Pandas, 2018). For the evaluation of our models, we used the LOSOCV method and we calculated the following metrics based on the confusion matrix (see Figure 9).

$$\textbf{\textit{Accuracy}} = \frac{TP+TN}{TP+FP+FN+TN}$$

$$\textbf{\textit{Precision}}\ \textit{(also known as positive predictive value)} = \frac{TP}{TP+FP}$$

$$\textbf{\textit{Recall}}\ \textit{(also known as sensitivity, hit rate, or true positive rate)} = \frac{TP}{TP+FN}$$

$$\textbf{\textit{F1- score}} = 2 * \frac{precision * recall}{precision + recall}$$

$$\textbf{\textit{Specificity}}\ \textit{(also known as true negative rate)} = \frac{TN}{TN+FP}$$

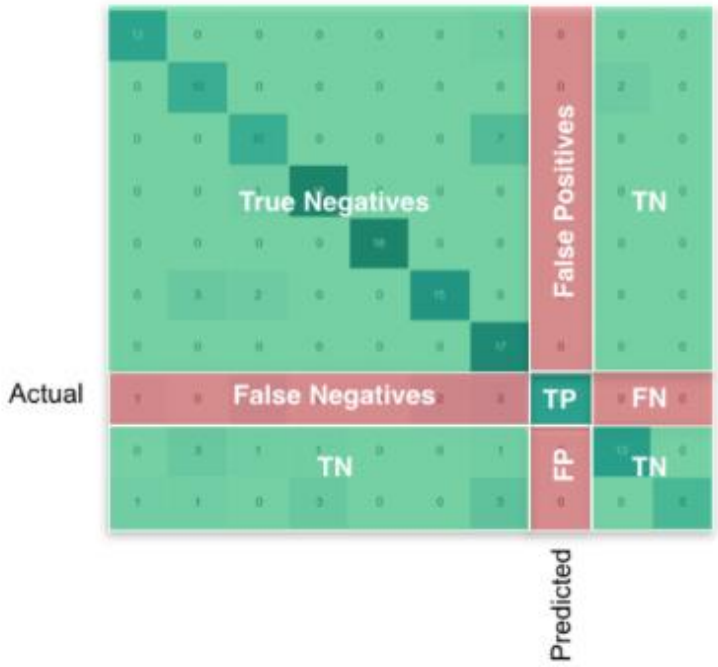


Figure 9: Example of a confusion matrix (Stackoverflow, 2018).

For detecting emotional and cognitive behaviours, we make use of the EyesWeb platform (Camurri, 2004) to detect head and torso movements and OpenFace (Baltrušaitis T., 2015) to detect the facial

expressions. Figure 10 shows an overview of the system. Facial display is a natural means of communicating emotions. Facial Action Coding System (FACS) is a system used to describe human facial movements by their appearance on the face (P. Ekman, 2002). Action Units (AUs) are the fundamental actions of individual muscles or groups of muscles. FACS has been widely used in recognition of basic emotions and complex psychological states. It is an objective method for quantifying facial movement in terms of component actions. We will be making use of AUs for analysing emotions.

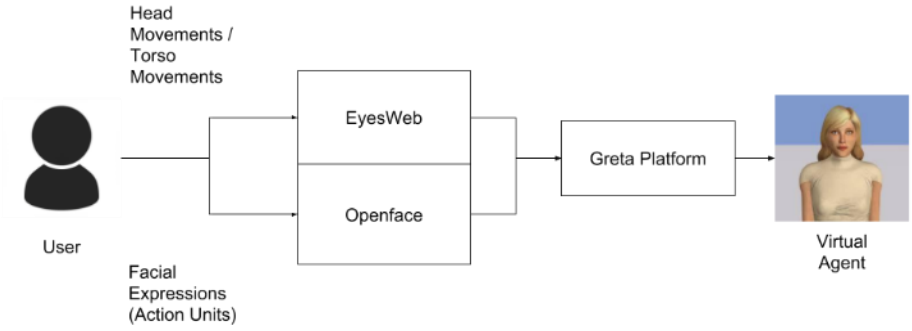


Figure 10: Overview of system for detecting non-verbal behaviours.

OpenFace is an open source tool intended for computer vision and machine learning researchers (Baltrušaitis T., 2015). The software is available for download at on GitHub[1]. It is capable of facial landmark detection, head pose estimation, facial action unit (AU) recognition, and eye-gaze estimation. To extract facial AUs that will serve as facial descriptors for our study set we make use of OpenFace.

The tool offers two kinds of scores for the AU (see Table 1):

1. Intensity: the intensity of 17 AUs on a continuous value scale from 1 (minimally present) to 5 (present at maximum intensity); a score of 0 indicates absence.
2. Presence: indicates the presence or absence of 18 AUs.

Table 1: List of AUs detected using OpenFace.

AU	Description
1	Inner Brow Raiser
2	Outer Brow Raiser
4	Brow Lowerer
5	Upper Lid Raiser
6	Cheek Raiser
7	Lid Tightener
9	Nose Wrinkler
10	Upper Lip Raiser
12	Lip Corner Puller
14	Dimpler
15	Lip Corner Depressor
17	Chin Raiser
20	Lip Stretcher

[1] https://github.com/TadasBaltrusaitis/OpenFace

23	Lip Tightener
25	Lips Part
26	Jaw Drop
45	Blink

EyesWeb XMI (Camurri, 2004) is a is an open software platform that supports the design and development of real-time multimodal systems and interfaces. EyesWeb is designed and developed by InfoMus Lab of University of Genova and available at *http://www.infomus.org/eyesweb_eng.php*. A wide number of input devices including motion capture systems, video cameras, game interfaces (e.g., Kinect, Wii), multichannel audio input (e.g. microphones), analogue inputs (e.g. for physiological signals) are supported by the platform. EyesWeb supports real-time synchronized recordings of multimodal channels, and includes several software libraries dedicated, for example, to expressive movement analysis (e.g., detecting social/affective signals).

The system uses EyesWeb XMI to analyse and record the user's non-verbal signals in real-time (e.g., facial expressions, torso movements, head gestures). In particular, EyesWeb manages the data coming from sensors (i.e., microphone, Kinect v2), extracting low-level signals (i.e., torso and head orientation) and computing mid-level features (i.e., body and head attention over time).

Also, it sends the data to a specific module (see Section 3.3.4) and collects the results of speech recognition and face engagement. Internally, it exploits OpenFace to extract facial Action Units (AUs) from Kinect's RGB information. Figure 12 depicts the user's analysis interface, developed in EyesWeb. On the left, the user's silhouette is extracted using Kinect's depth data. The two red bars in the middle indicate that the user is looking at the screen, with both her torso (left bar) and head (right bar). Audio intensity is low (volume meter on the right), that is, the user is not speaking. Finally, full-body engagement level (between 0 and 5) is represented by the green bar on the right. Details about full-body engagement computation are provided later in this document in Section 3.3.4.

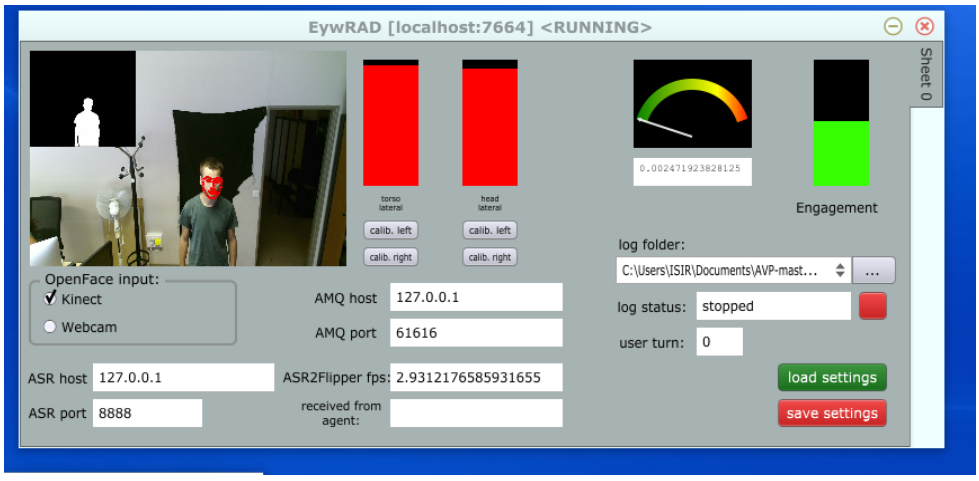


Figure 11: EyesWeb Interface.

3.3.1 Physical Behaviour Model

The model for detecting short-term physical behaviour has been divided into two parts. For the first part, we focus on detecting steps based on accelerometer data, while for the second part we develop a model for detecting everyday activities (walking, being in a vehicle, biking, tilting or remaining still) using

accelerometer and GPS data. In addition, we used the Google Recognition API plugin (Activity Recognition API, 2018) as part of the validation of our own model for the activity recognition part.

For the steps counter model, we calculated the magnitude value of 3-axial accelerometer raw data (50Hz sampling frequency) in order to make the model orientation independent. Then, we applied a lowpass filter with a cut-off frequency equal to 3.667 dB (we tested different values and we chose this specific cut-off as the most optimal). The filter passes signals with a frequency lower than the cut-off frequency and attenuates signals with frequencies higher than the cut-off frequency. After removing short-term fluctuations, we counted the peaks of the magnitude signal above a certain threshold, where each peak equals to one step. This threshold is equal to 12 $(m/s)^2$ and was selected among other values as the most optimal one for this steps counter model. In the following figures (see Figure 13, Figure 14 & Figure 15), we present the step counter model for a different range of short-term periods. It is worth mentioning, that the actual steps are defined based on a pedometer, while the detected steps are calculated based on our model.

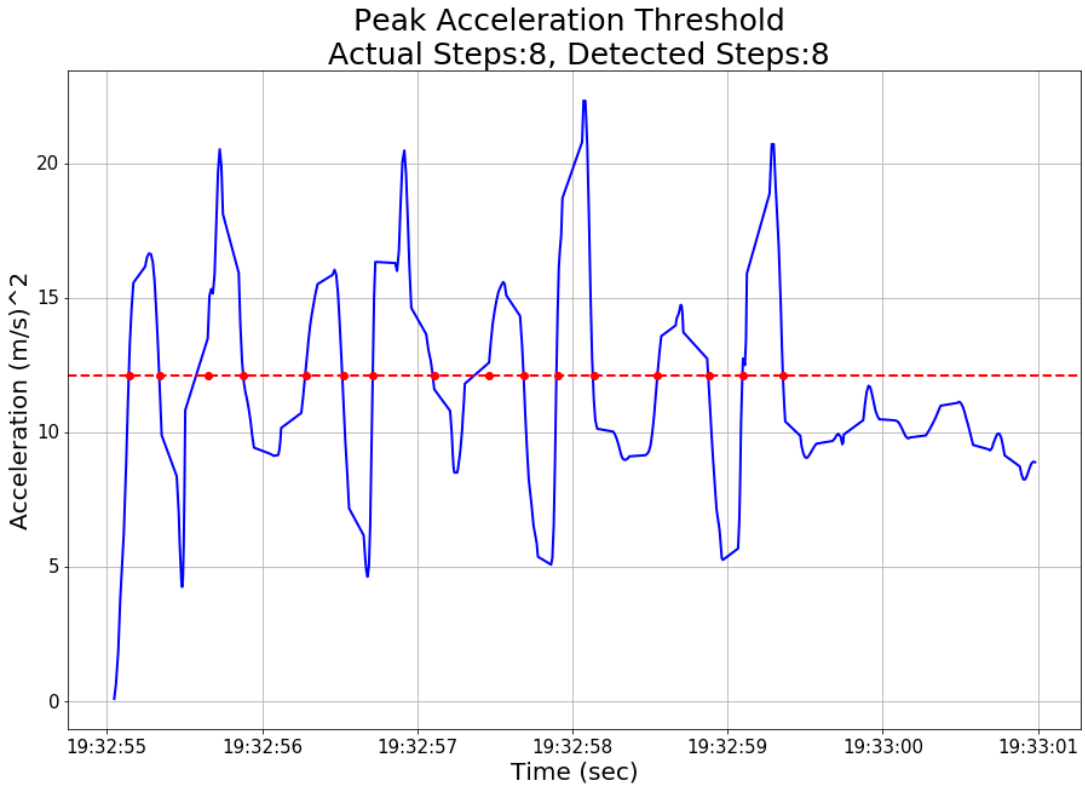


Figure 12: Steps Counter model for 5 seconds.

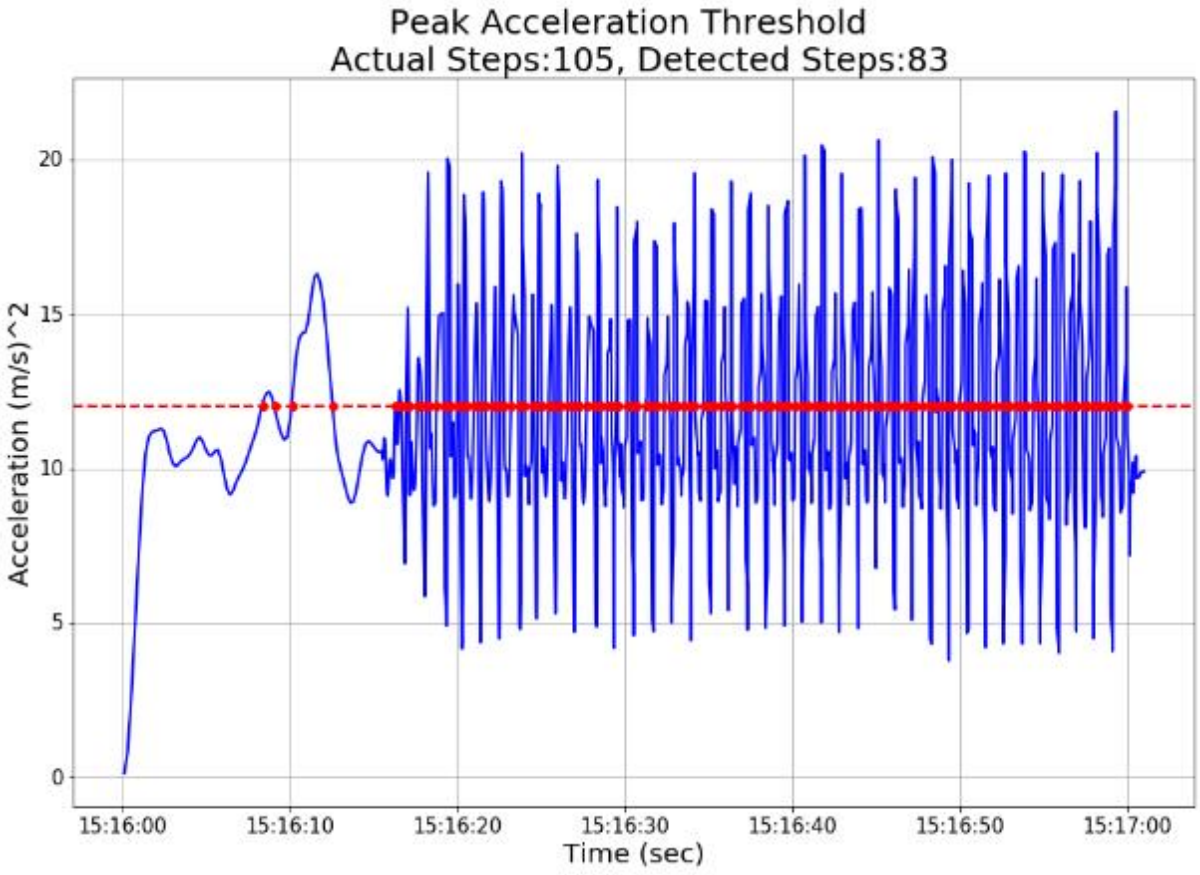


Figure 13: Steps Counter model for 60 seconds.

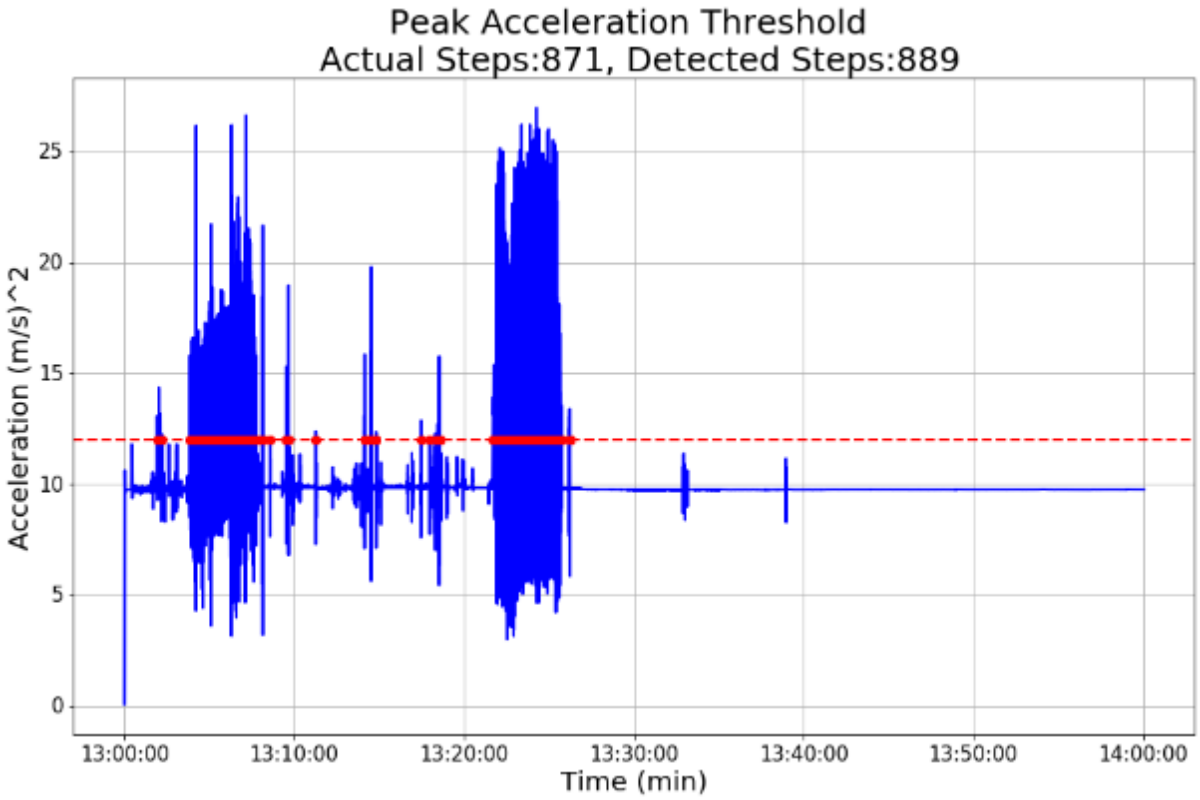


Figure 14: Steps Counter model for 1 hour.

For the activity recognition model**,** we processed accelerometer data in a 2 seconds window size with 1 second overlap. Overall, small window segments with 50% overlap are mainly used in activity recognition systems through smartphone data (Labrador, 2013) (Y. P. Chen, 2008). Based on this window segment, we calculated 52 features in time and frequency domain (see Table 1). Features related to body posture measurements are represented by the mean, median, area, and meandistance; features related to motion shape measurements are represented by absmean, absarea, magnitude, entropy, skewness and kurtosis; features related to motion variation measurements are represented by variance, std, amplituderange and interquartilerange; and features related to motion spectral content measurements are represented by signalpower and fftfreq (Tapia, 2008).

Table 2: An overview of the calculated features.

Feature Name	Number of Features	Domain	Description
mean (x,y,z)	3	time	computes the mean; the average of all sample values in a sample window
absmean (x,y,z)	3	time	computes the absolute mean; the absolute average of all sample values in a sample window
median (x,y,z)	3	time	computes the median; the value that divides the higher half from the lower half in a sample window
variance (x,y,z)	3	time	computes the variance; the average of the squared differences of the sample values from the mean in a sample window
std (x,y,z)	3	time	computes the standard deviation; the square root of variance in a sample window
max (x,y,z)	3	time	computes the min; the lowest number of all sample values in a sample window
min (x,y,z)	3	time	computes the max; the highest number of all sample values in a sample window
magnitude	1	time	computes the magnitude; by adding each one of the squared axes in a sample window, and calculating the square root of the sum
skewness (x,y,z)	3	time	computes the skewness; the asymmetry of the distribution of the sample values around the mean in a sample window
kurtosis (x,y,z)	3	time	computes the kurtosis; the shape description of the distribution of the sample values in a sample window
meandistance (x-y, x-z, y-z)	3	time	computes the mean distance; the differences between the mean values of the x-y, x-z and y-z in a sample window
amplituderange (x,y,z)	3	time	computes the amplitude range; the difference between the maximum and minimum sample values in a sample window

interquartilerange (x,y,z)	3	time	computes inter quartile range; the difference between quartiles[2] Q3 and Q1, and describes the dispersion of the acceleration signal
area (x,y,z)	3	time	computes the area; the sum of the sample values in a sample window
absarea (x,y,z)	3	time	computes the absolute area; the sum of the absolute sample values in a sample window
entropy (x,y,z)	3	frequency	computes the entropy; the degree of distortion in a sample window (discriminate activities that have the same Power Spectral Density[3] but different patterns of movement)
signalpower (x,y,z)	3	frequency	computes the signal power; the non-normalized sum of the Power Spectral Density in a sample window
fftfreq (x,y,z)	3	frequency	computes the Fast Fourier Transform peaks; the frequency related to the highest computed Power Spectral Density in a sample window

After extracting features, the phase of feature selection aims to filter the feature set and remove any redundant features, reducing the dimensionality and improving the overall classification performance. At first, highly correlated features are removed based on the Pearson's correlation coefficient, which measures a linear correlation between features (scipy.stats.pearsonr, 2018). For each feature, the correlation coefficients are calculated and ranked according to the other features, starting from the lowest scored feature. Hence, if the selected feature has a correlation coefficient higher than the threshold (equivalent to an absolute value of 0.80) with at least one feature, then the features are removed. For instance, the 'mean' features are correlated with the 'median' features, and thus, only one of these two is kept. Secondly, non-informative features, with low information gain, are removed. Each feature is ranked based on the gain ratio, which is measured respecting the contribution of each feature to the accurate prediction. For this phase, low information gain is based on linear estimators, such as linear SVC (Support Vector Classifier), by removing unnecessary features (scikit-learn, 1.13. Feature selection, 2018). Overall, 52 features are extracted and then reduced to 23 after the feature selection phase.

After data processing, the dataset was divided into training and test sets using the Leave-One-Subject-Out Cross Validation (LOSOCV) method. This method is used in order to avoid overfitting, by excluding data from subjects that were used for training the classifier (learning model) and include only unseen data for validating our model. The model was trained based on two classification algorithms, Support Vector Machine and Random Forest, which are recommended in activity recognition problems (M. Gjoreski, 2016).

3.3.2 Social Behaviour Model

The model for detecting user's social behaviour aims to answer if the user interacts with other individuals and has a socially active or isolate life. Thus, different types of smartphone data are processed in order

[2] Quartiles are calculated by partitioning the sample values of a sample window into four quarters, each one contains 25% of data, with Q1=25%, Q2=50% and Q3=75% (M. Shoaib, 2015).
[3] Power Spectral Density (PSD): the squared sum of its spectral coefficients, normalized by the number of the window slide (F. Attal, 2015).

to estimate the level of being socially active. The signal strength of Bluetooth visible devices is translated to social interaction. The model estimates that if a user is surrounded closely by Bluetooth devices there might be individuals to interact with. The condition for that is the RSSI values to range from -70 to 0 dBm. Furthermore, the number of performed incoming and outcoming phone calls is also used to measure social interaction. The model checks every minute if at least one call is performed for more than 10 seconds (social interaction). Similarly, the number of received and sent text SMS messages is translated to social interaction. The model checks every minute if at least one SMS is received and one SMS is sent (social interaction). Moreover, ambient noise is recorded in order to evaluate if a user has a conversation with others or if the user is in a noisy environment (social interaction). The condition for that is the decibels values to be more than 30 dB and the RMS to be more than 150 dB. Finally, the model detects the location of the user through the Google API activity recognition, and estimates that if a user is taking the bus, there will be an interaction with the bus driver or with other passengers. For any other case, the model estimates that the user is socially isolated. Consequently, if there is at least one sensor that indicates that the user is socially active per minute, then the label for this minute is social active. The approach of the social behaviour model is illustrated in the following figure (see Figure 16).

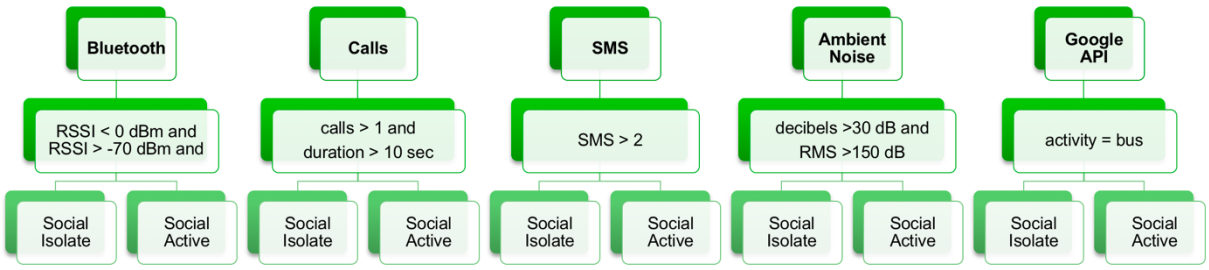


Figure 15: Illustration of the layered social behaviour model.

3.3.3 Emotional and Cognitive Behaviour Model

We constructed neural networks to model the emotion level of the user in human-agent interaction using Keras toolkit with TensorFlow. Our model was designed to use the actions units, the head rotation and the conversational state of the interaction to predict the engagement, arousal and valence level of the user.

The model uses softmax as activation function which is used for multiclass classification. We trained the model via 10-fold cross-validation. Within each fold, 90% of data was used as the training set and 10% as the test set. As our data was imbalanced cf. Sec. 4, we used SMOTE (Synthetic Minority Over-sampling Technique) technique to balance the data. A categorical cross-entropy was used to compute the loss of the model. Finally, a dropout is performed to prevent any over-fitting.

The prediction model takes as input the action units, the head rotation of the user and the conversational state of the interaction during the last 30 frames and predicts the emotion level (in total, we have five level) of the user for the next frame. The input is:

- Intensity (from 0 to 5) of 17 AUs (AU01, AU02, AU04, AU05, AU06, AU07, AU09, AU10, AU12, AU14, AU15, AU17, AU20, AU23, AU25, AU26, AU45),
- Presence (0 absent, 1 present) of 18 AUs (AU01, AU02, AU04, AU05, AU06, AU07, AU09, AU10, AU12, AU14, AU15, AU17, AU20, AU23, AU25, AU26, AU28, AU45)
- Head rotation (rotation is in radians around X, Y, Z axes)
- Conversation state of the interaction cf. table 2.

These inputs are detected in real time using EyesWeb. We evaluate our model using two corpora, RECOLA (Ringeval F., 2013) and NoXi (Cafaro, 2017), to predict respectively arousal and valence level using RECOLA and to model and predict engagement level of both expert and novice from NoXi corpus cf. Sec 4.

Table 3: The conversation states for agent and user.

Conversation State	Meaning
NONE	Silence
BOTH	Both agent and user are speaking
Agent	Agent is speaking
User	User is speaking

Starting from the body low-level features extracted by EyesWeb and the engagement level predicted by the model described above, we defined a Full-body Engagement Model, based on the works of Corrigan et al. (Corrigan, 2016) and Sidner & Dzikovska (Sidner C. L., 2005) . In our model, we focus on the work of Corrigan et al. (Corrigan, 2016), illustrating how engagement can be expressed by different types of low-level signals:

- Attention, primary level of engagement, can be expressed by a person by continuously gazing at relevant objects/persons during the interaction. The more a person continuously focuses her attention for a relevant object/person, the more engaged she is.
- According to (Corrigan, 2016), "frowning may indicate effortful processing suggesting high levels of cognitive engagement". The same study also refers to signals such as "looking for a brief interval outside the scene" as indicators of cognitive engagement.
- Finally, the engagement has also an affective component: smiling could indicate that a person is enjoying interaction, while some postures (e.g., crossed arms, hands in pockets) or posture shifts can indicate a lack of engagement.

In our model of engagement, we consider all the above signals, except those related to posture. While the affective and cognitive components are considered by the prediction model (e.g., signals like frowning, looking away, smiling), the attention signals are analysed by EyesWeb. For example, if the user is looking at the agent (with both her face and her torso) then the attention level increases and we apply a bonus to the output of the prediction model. All these indices are collected throughout the entire speaking turn, that is, from the moment the agent starts to speak to the moment the user stops to speak (or, if the user does not respond, until a 1500 ms silence threshold is passed).

We implemented the above model in an EyesWeb patch (i.e., a program written with the visual programming language of EyesWeb). Below, we describe the steps of computation of the patch.

```
Input: Kinect RGB video and depth map
Output: Full-Body Engagement (FBE)
Parameters: angle_threshold = 15 degrees

before beginning the next speaking turn:
    attention_head = 0, attention_body = 0; PEng = 0;
while(speaking turn is active){
    AUs=OpenFace(Kinect RGB) // call OpenFace to extract face AUs from KinectRGB
    head_angle=GetPose(Kinect depth map).Head.y
    PEng=PEng+EngagementPredictionModel(AUs,head_angle);
```

```
    body_angle=GetPose(Kinect depth map).Shoulders.y
    if (head_angle<angle_threshold) increase(attention_head, 1);
    if (body_angle<angle_threshold) increase(attention_body, 1);
}

PEng=PEng/speaking_turn_length; //mean predicted engagement is an integer value in
[0,4]
AH=attention_head/speaking_turn_length; // this is a real value in [0,1]
AB=attention_body/speaking_turn_length; // this is a real value in [0,1]
Att=Rescale(max(AH,AB),-1,1); // rescale max(AH,AB) in [-1,1]
FBE=Limit(FE+Att,0,5); // this is a real value in [0,5]
```

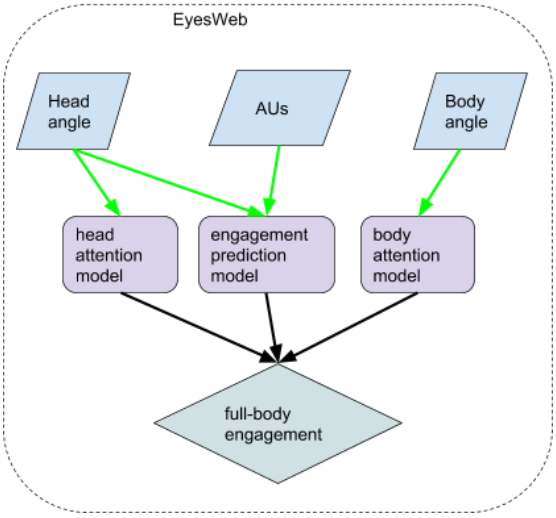


Figure 16: Full body engagement model.

EyesWeb communicates with the prediction model (PM) described in the beginning of Sec. 3.3.3, through a TCP connection. The PM sets up a TCP server on port 8890, while EyesWeb implements a parallel thread to send and receive data to the server. At each video frame, EyesWeb calls the OpenFace API to get the user's facial Action Units configuration and head rotation. The PM predicts user's engagement and provides it to EyesWeb as an integer value in the interval [0,4]. The following string is an example of the AUs and head data sent from EyesWeb to the PM:

```
0.410933;0.39483;0.372371;0.338012;0;0;0.493247;0;0;0.157038;0.20112;0.409727;0.378
909;0.563469;0.543542;0.480755;0.613564;0;0;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;
```

The string format is the following:

```
AU01_r;AU02_r;AU04_r;AU05_r;AU06_r;AU07_r;AU09_r;AU10_r;AU12_r;AU14_r;AU15_r;AU17_r
;AU20_r;AU23_r;AU25_r;AU26_r;AU45_r;AU01_c;AU02_c;AU04_c;AU05_c;AU06_c;AU07_c;AU09_
c;AU10_c;AU12_c;AU14_c;AU15_c;AU17_c;AU20_c;AU23_c;AU25_c;AU26_c;AU28_c;AU45_c;Hx;H
y;Hz;S;
```

where:

- AUxx_r is the intensity (in [0,5]) of AU xx
- AUxx_c is the presence (in [0,1]) of AU xx
- Hx,Hy,Hz is the head rotation
- S is a Boolean flag, where 0 means that the user is not speaking, while 1 means that the user is speaking

After receiving the predicted engagement from PM, EyesWeb runs the Full-Body Engagement algorithm and generates the following XML code, based on the ARIAVALUSPA format, described in https://github.com/ARIA-VALUSPA/AVP/wiki/Documentation#ssi:

```
<?xml version="1.0" ?>
<user>
<arousal short="0.0" long="0.0" diff="0.0" />
<valence short="0.0" long="0.0" diff="0.0" />
<engagement short="2.000000" long="" diff="0.0"  isnew = "0"/>
<gender male = "0.5" female = "0.5" />
<age child = "0.25" youth = "0.25" adult = "0.25" senior = "0.25" />
<head horizontal = "0.0" vertical = "0.0" activity = "0.0" />
<voice active = "0.026428" />
<speech time = "0" dur = "5030" isnew = "1">
   <ASR_output>
   <idur>1.200</idur>
   <partial>False</partial>
   <transcriptions><id>0</id><text>Hello, I am Alice.</text>
       <nwords>4</nwords>
   </transcriptions>
   <language>en</language>
   <mode>utt</mode>
   <rdur>3.78344</rdur>
   <nbest>1</nbest>
   </ASR_output>
   </speech>
<face id = "1"/>
<emotions neutral = "0.0" anger = "0.0" disgust = "0.0" fear = "0.0" happiness =
"0.0" sadness = "0.0" surprise = "0.0" />
</user>
```

As described previously, the Full-Body Engagement algorithm needs to know when each speaking turn starts (i.e., when the agent starts to speak) and ends (i.e., when the user ends to speak). This information is managed by the Dialog Manager Flipper developed by University of Twente, which sends messages through network to EyesWeb. To do that, we exploit ActiveMQ communication, on which Greta/VIB is based. In particular, we defined a new ActiveMQ topic called "DialogTurn" on the ActiveMQ board managing the communication between Greta/VIB modules, on which we send and receive the following messages:

- agent_start: it is sent when the agents starts its speaking turn
- agent_end: it is sent when the agent ends its speaking turn
- user_end: it is sent when the user ends her speaking turn (i.e., we consider that the user starts to speak as soon as the agent ends its turn of conversation, so we do not send another message to indicate it)

4 Evaluation

4.1 Physical Behaviour

4.1.1 Experimental Setup

For the data acquisition phase, we recruited 8 healthy subjects and we asked them to perform a series of consecutive activities for around 30 minutes, voluntarily. Participants were master students and employees at the University of Twente (older than 18 years old), and they were recruited individually through emails and printed information brochures. They were asked to follow the researcher's instructions (controlled experiment) and use a smartphone device for acquiring different types of data, with respect to user's privacy. Furthermore, the subjects wore a GoPro camera on their chest for taking pictures every 5 seconds. The pictures were used in a later phase as ground truth labels for the activity classification. The experiment took place in indoors and outdoors areas of the University of Twente.

The participants were asked to perform the activities based on the following protocol.

1. Sitting and being still (5 minutes)
2. Standing and being still (5 minutes)
3. Tilting: Sitting-to-Standing and Standing-to-Sitting (1 minute)
4. Walking (10 minutes)
5. Cycling (5 minutes)
6. Taking the Bus (5 minutes)

4.1.2 Results

For the evaluation of the steps counter model, we tested different time frames and we concluded that counting steps for one hour performs more accurate compared to smaller periods of time. In order to reduce the detection of false peaks due to a random movement of the device, we made our model less sensitive to noise. However, some actual steps that are solely performed with a magnitude < 12 $(m/s)^2$ are not considered as significant peaks. In Figure 18, we evaluate our model in a daily life scenario, where we plot the actual versus the detected number of steps for one hour. Similarly, in Figure 19, we depict the error rate (1 minus accuracy score) of the model for the same scenario.

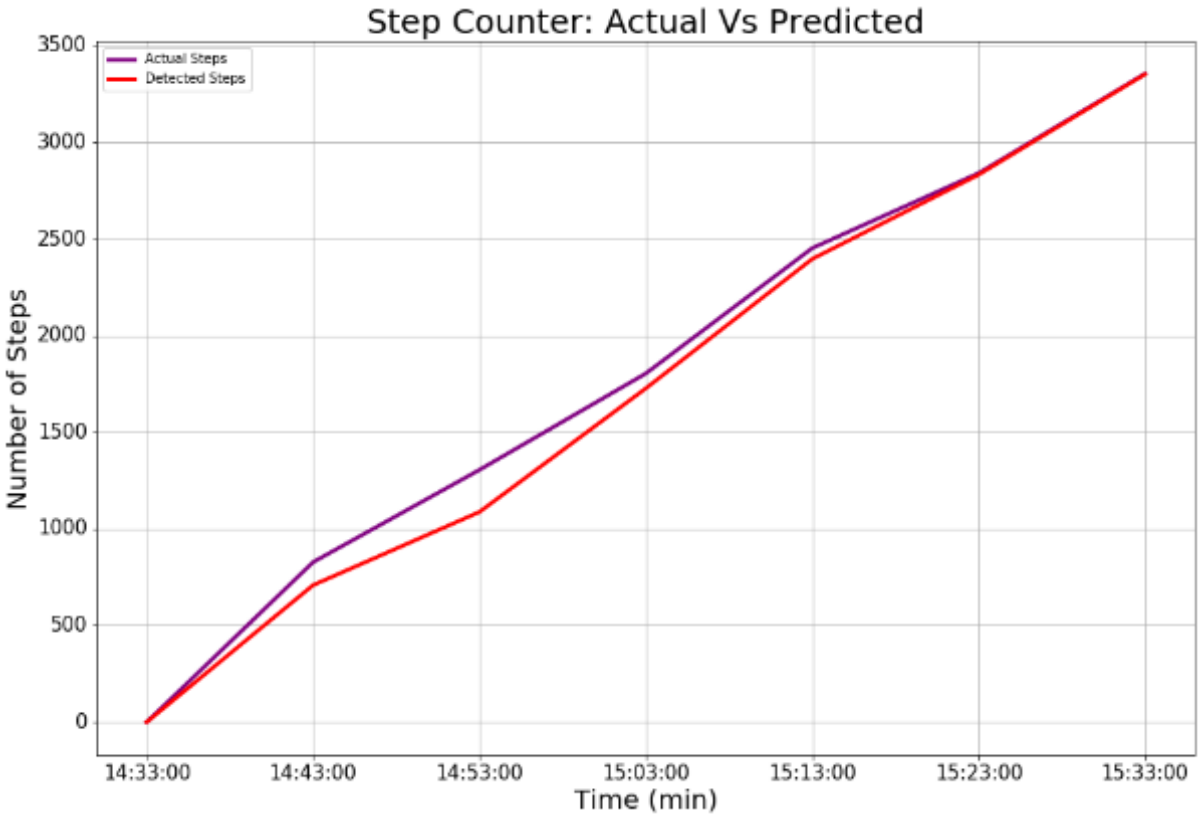


Figure 17: Evaluation of steps counter model based on actual and detected steps during a one hour scenario.

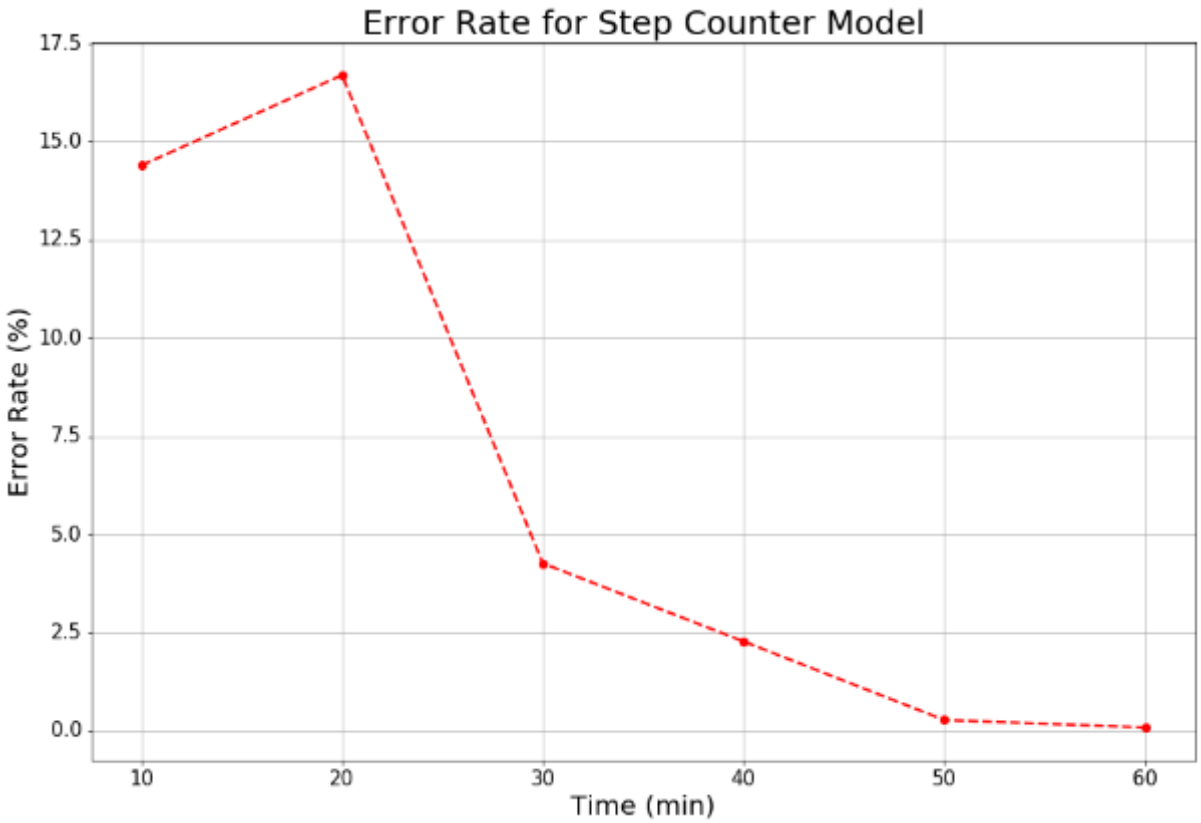


Figure 18: Error rate for the steps counter model during a one-hour scenario.

For the evaluation of the activity recognition model, we examined two different approaches. For the first one, we tried to detect 5 activities based on processed accelerometer data. More precisely, we evaluated the performance based on two classification algorithms; the SVM and the Random Forest. Overall, the Random Forest performed slightly better and was selected for the classification model (see also Figure 20).

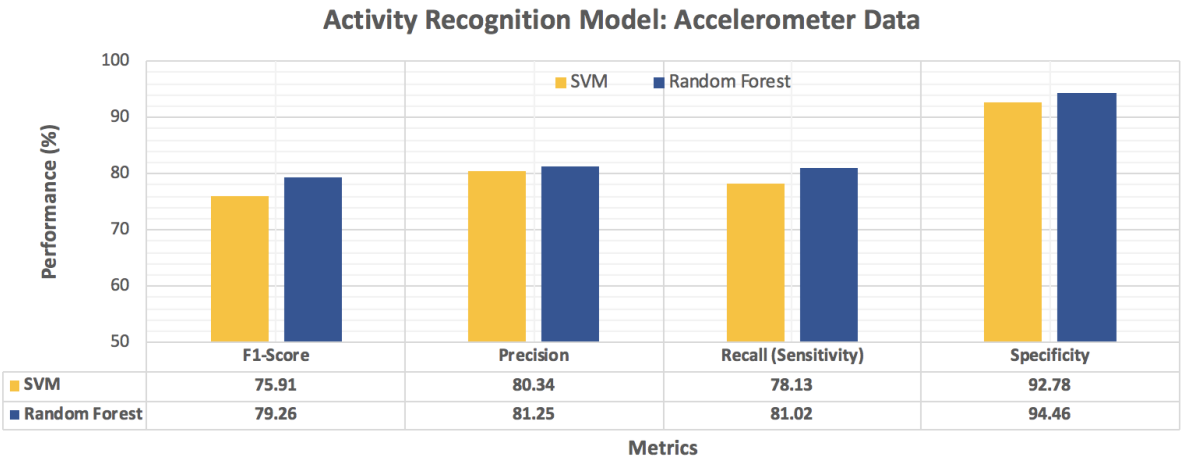


Figure 19: Evaluation metrics for the activity recognition model using accelerometer data.

In order to enhance the classification performance, we used the GridSearch optimization in order to find the most optimal parameters for the Random Forest algorithm. In particular, we used the following parameters and the F1-score was increased to 83.43% (see Figure 21).

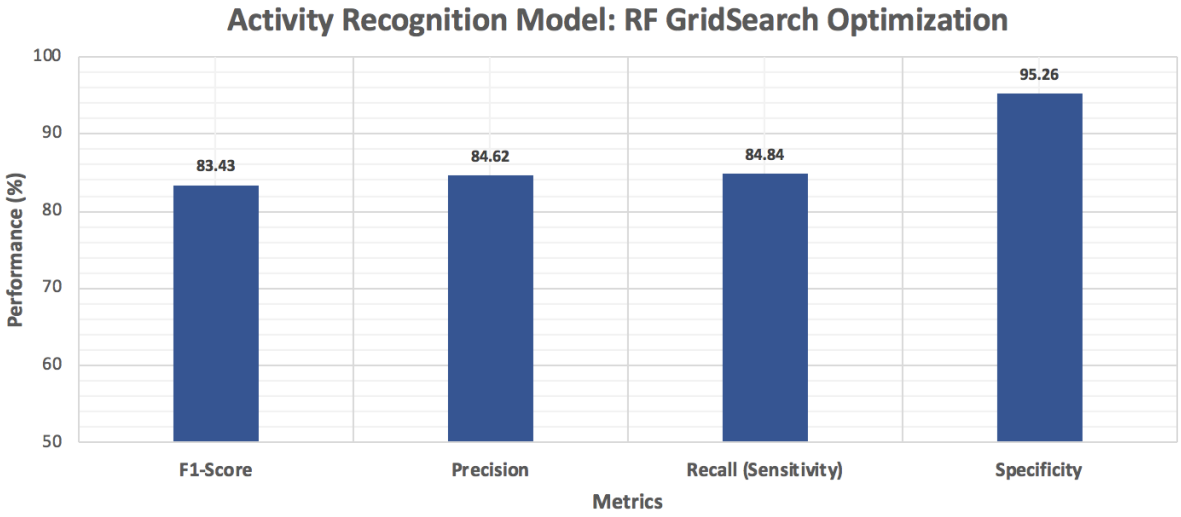


Figure 20: Evaluation metrics for the activity recognition model using the GridSearch optimization for the Random Forest algorithm.

For the second approach, we combined accelerometer and GPS data in order to define the activity based on user's location. In particular, we used 23 extracted features (after feature selection) from accelerometer data and the latitude and longitude features from GPS data. However, this approach using GPS data did not enhance the classification performance (see Figure 22).

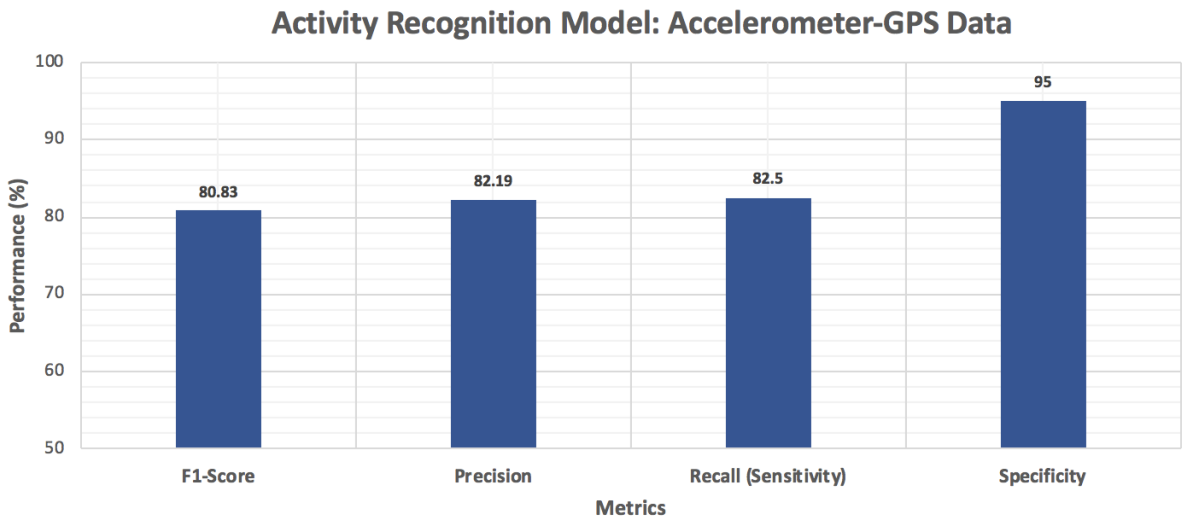


Figure 21: Evaluation metrics for the activity recognition model using accelerometer and GPS data.

Overall, our prediction model receives 83.43% F1-score, 84.62% precision, 84.84% recall and 95.26% specificity, using the optimized parameters of the Random Forest classifier. Based on the following confusion matrix (see Figure 23), we can see the actual versus predicted classes for the performed activities. The true positive score for the activity taking the bus is 83%, for the activity cycling is 41%, for the activity walking is 92%, for the activity being still (sitting and standing) is 91%, and for the activity tilting is 12%. It is worth mentioning that some activities are misclassified. For instance the activity cycling is mainly misclassified with taking the bus, while the activity tilting is misclassified with the activity being still.

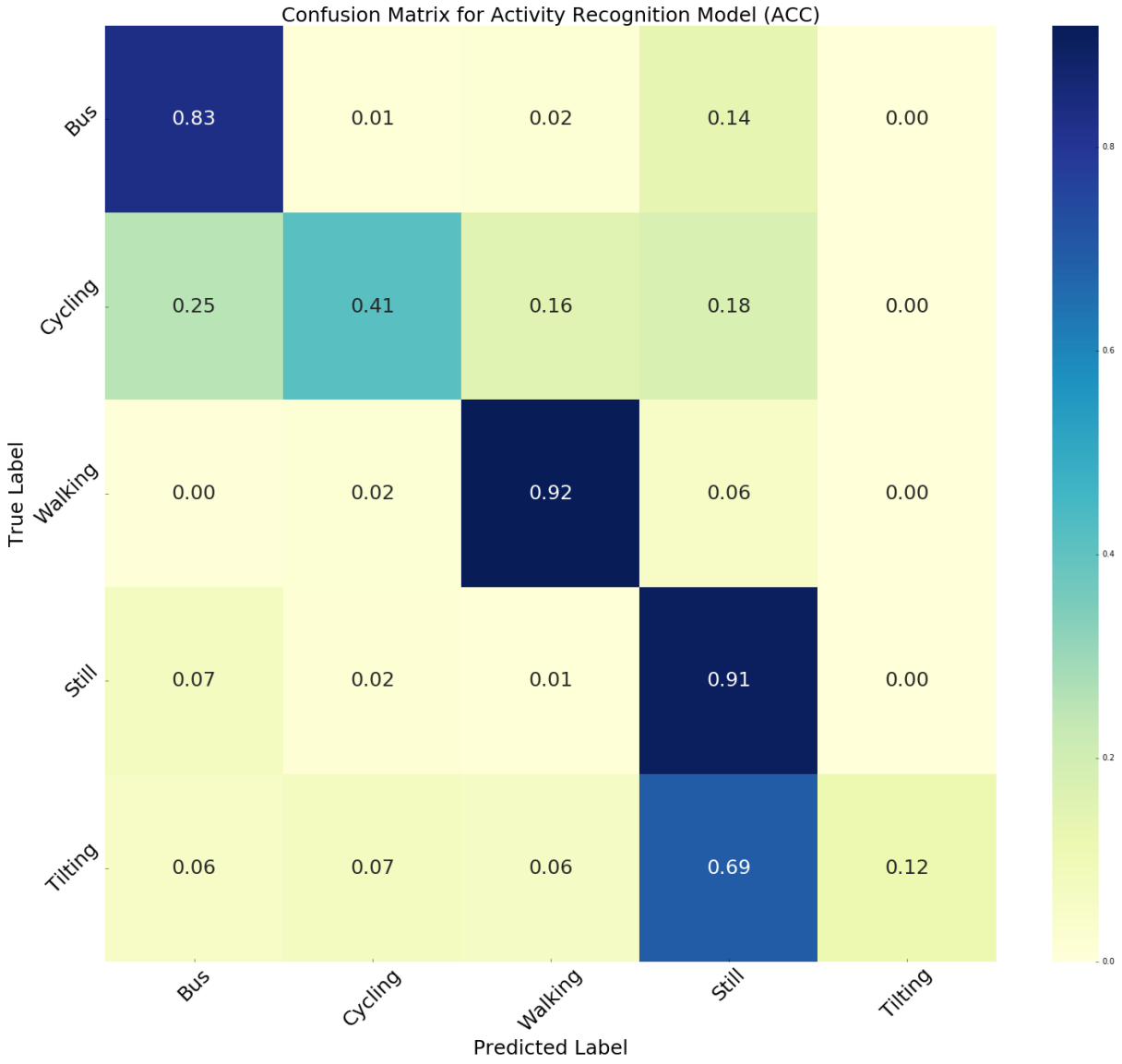


Figure 22: Confusion matrix for the activity recognition model using accelerometer data.

For the validation of our activity recognition model**,** we compared the achieved classification performance with the one from the Google API Activity Recognition (plugin). In particular, we tested two different models. The prediction model A consists of all the predicted classes, while the prediction model B consists of predicted classes only with a confidence score above 70%. The confidence score refers to how accurately the different classes have been predicted. For instance, when the Google classification algorithm predicts a class (for a performed activity based on 1-minute frame) with a low accuracy score (the class is misclassified with other classes and is prone to uncertainty), this predicted class receives a low confidence score. In Figure 24, we present the performance for the two models and we show that the model B performs slightly better compared to model A.

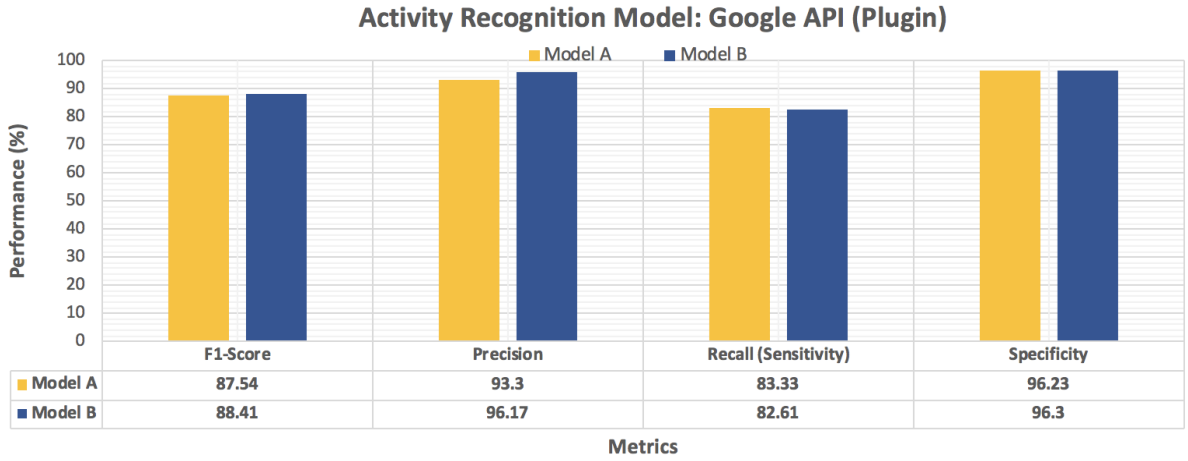


Figure 23: Evaluation of the Google API model.

The Google prediction model A receives 87.54% F1-score, 93.3% precision, 83.33% recall and 96.23% specificity. Based on the following confusion matrix (see Figure 24), we can see the actual versus predicted classes for the performed activities. The true positive score for the activity taking the bus is 74%, for the activity cycling is 86%, for the activity walking is 92%, for the activity being still (sitting and standing) is 82%, and for the activity tilting is 50%. It is worth mentioning that the activity tilting is misclassified with the activity being still.

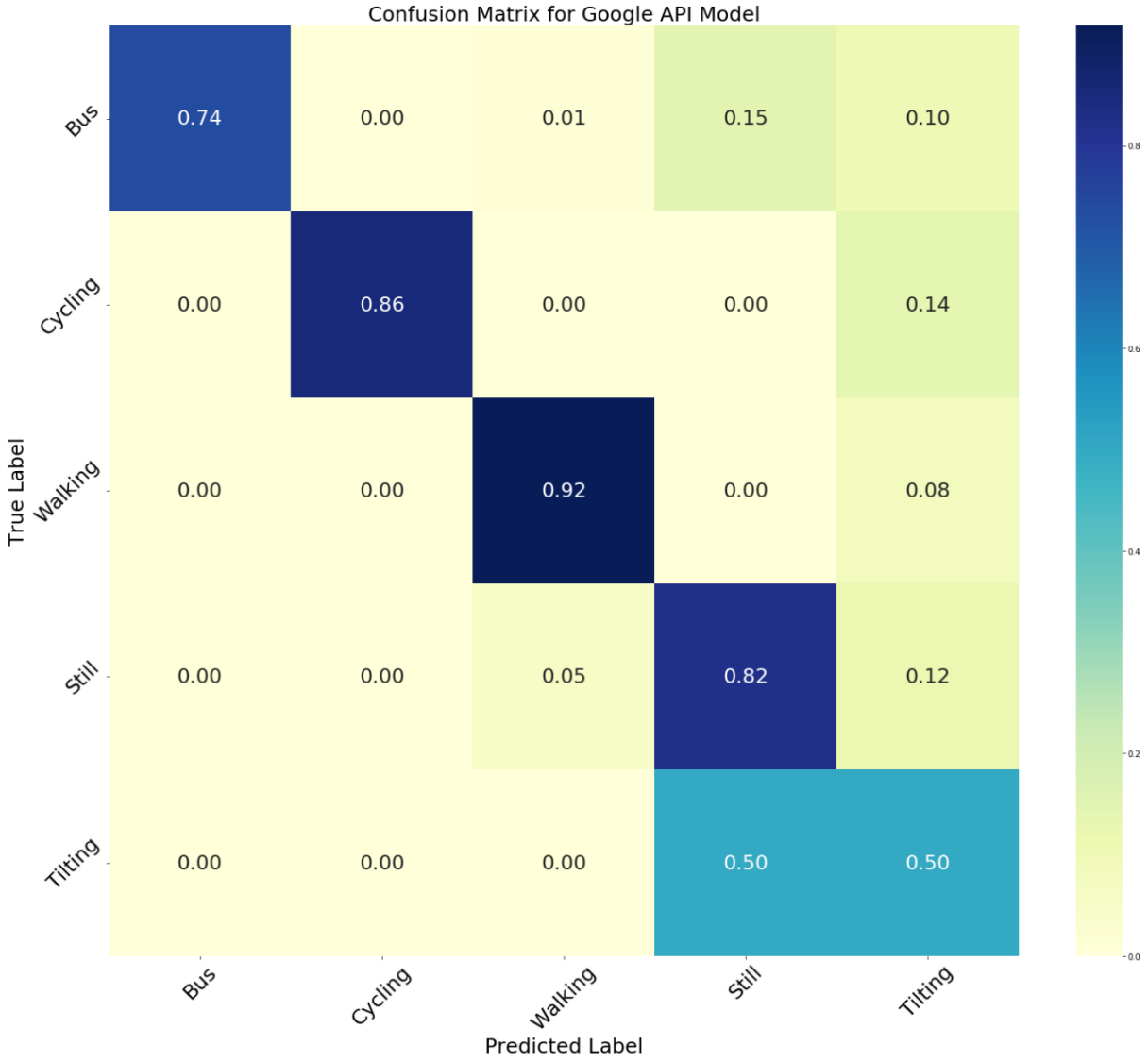


Figure 24: Confusion matrix for the Google API Activity Recognition (Model A).

The Google prediction model B receives an 88.41% F1-score, 96.17% precision, 82.61% recall and 96.3% specificity. Based on the following confusion matrix (see Figure 26), we can see the actual versus predicted classes for the performed activities. The true positive score for the activity taking the bus is 71%, for the activity cycling is 84%, for the activity walking is 91%, for the activity being still (sitting and standing) is 76%, and for the activity tilting is 100%. It is worth mentioning that the activity tilting is predicted accurately. However, this prediction model is prone to overfitting with the activity tilting. Overall, it is clear that the Google API prediction model B achieves the best score and performs significantly better compared to our own activity recognition model, and especially for the activities cycling and tilting. Consequently, we decided to use the Google Model B for the detection of short-term physical behaviours.

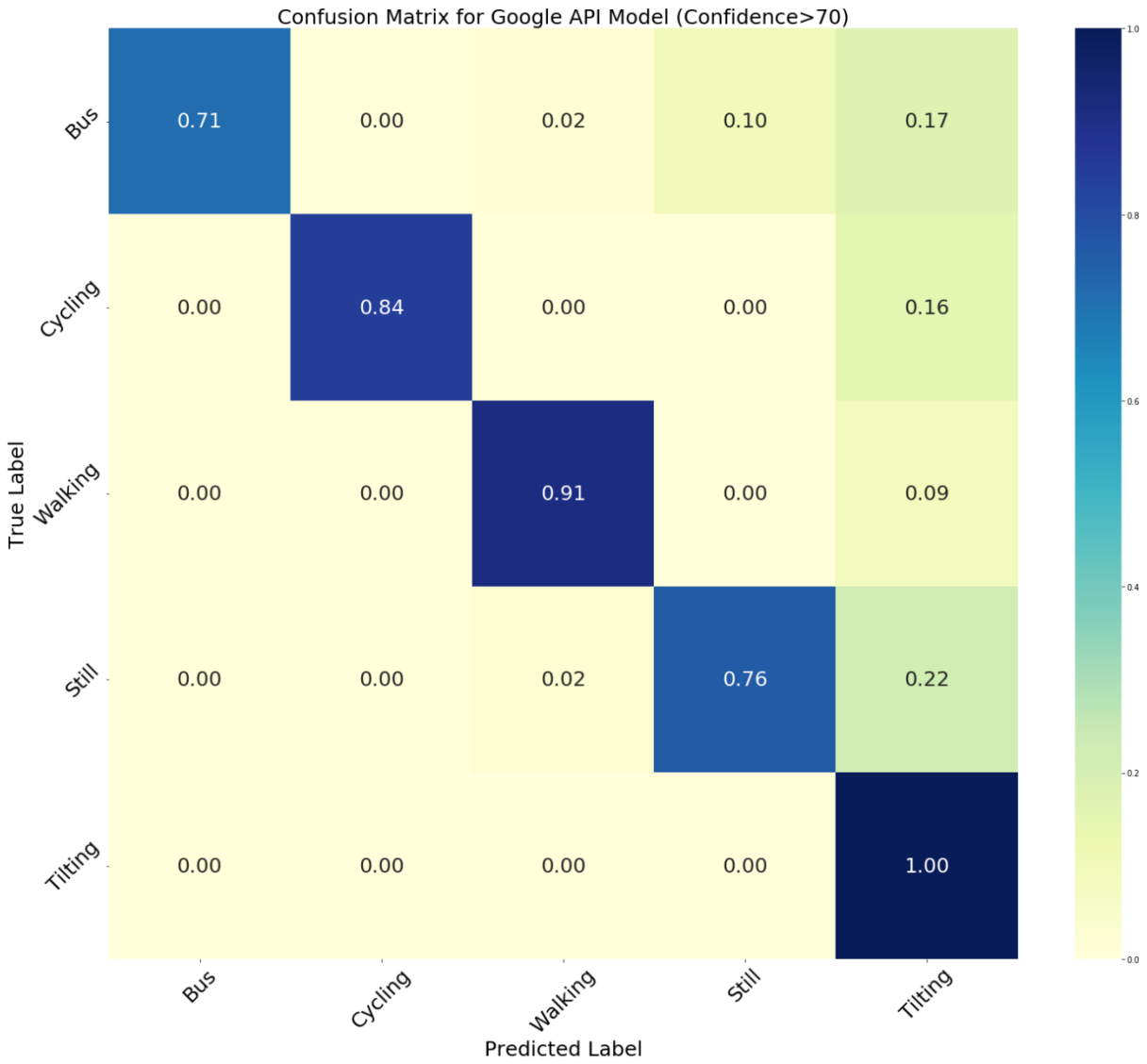


Figure 25: Confusion matrix for the Google API Activity Recognition (Model B).

4.2 Social Behaviour

4.2.1 Experimental Setup

For the data acquisition phase, we recruited 8 healthy subjects and we asked them to perform a series of consecutive tasks, voluntarily, for around 30 minutes. Participants were master students and employees at the University of Twente (older than 18 years old), and they were recruited individually through emails and printed information brochures. They were asked to follow the researcher's instructions (controlled experiment) and use a smartphone device for acquiring different types of data, with respect to user's privacy. Furthermore, the subjects wore a GoPro camera on their chest for taking pictures every 5 seconds. The pictures were used in a later phase as ground truth labels for binary classification (social active or social isolate). The experiment took place in indoors and outdoors areas of the University of Twente.

The participants were asked to perform different tasks based on the following protocol.

1. Being alone and reading a paper (5 minutes)
2. Being alone and playing a game in smartphone (5 minutes)
3. Being alone and using a smartphone/watch a video on YouTube (5 minutes)
4. Interacting with the researcher (5 minutes)
5. Sending/Receiving text SMS messages (2 minutes)
6. Incoming/Outcoming phone calls (2 minutes)
7. Visiting a public location (10 minutes)

4.2.2 Results

The prediction model for the social behaviour uses a customized binary classification and detects if the user is socially active or isolate. In particular, the model receives a 79.75% F1-score, 81.36% precision, 82.03% recall and 67.64% specificity (see Figure 27). Based on the following confusion matrix (see Figure 28), we can see the actual versus predicted classes for the performed activities. The true positive score for isolation is 39% and for interaction is 96%. It is worth mentioning that the model detects accurately when a subject interacts with others. However, the class being social isolate is misclassified.

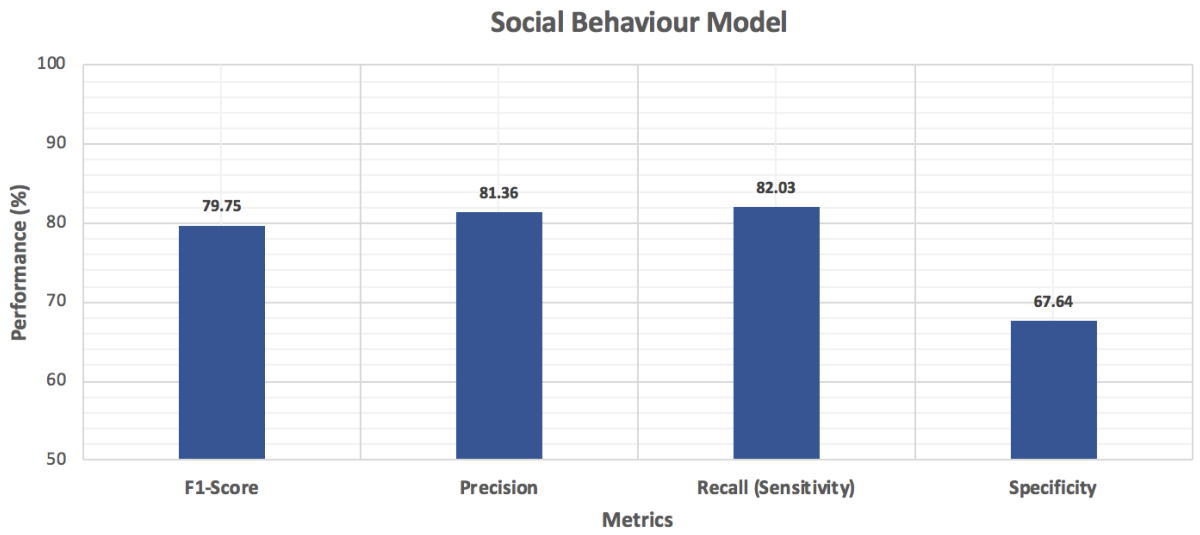


Figure 26: Evaluation of the Social Behaviour model.

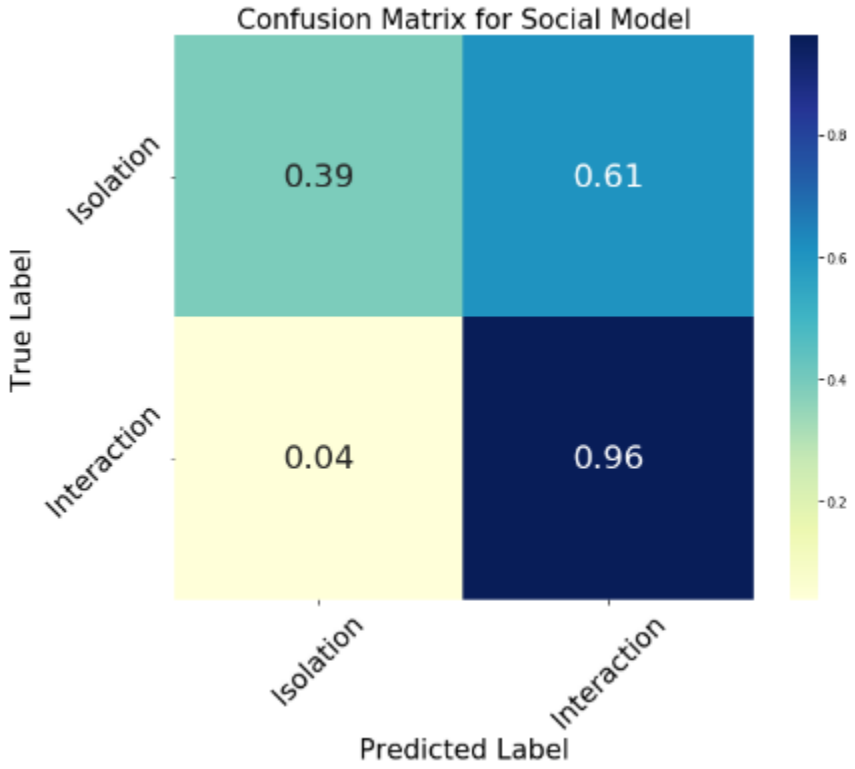


Figure 27: Confusion matrix for the Social Behaviour model.

4.3 Emotional Behaviour

4.3.1 Experimental Setup

To train our model we make use of the RECOLA database. The REmote COLlaborative and Affective interactions (RECOLA) data set is a multimodal emotional database of spontaneous interactions in French which consists of audio, visual, electro-cardiogram (ECG) and electro-dermal (EDA) data recorded continuously and synchronously (Ringeval F., 2013). It is publicly available[4]. Spontaneous interactions from 53 participants were recorded while solving a collaborative task: "Winter Survival Task" as dyadic teams. The recordings which are 9.5 hours long were made in isolated rooms and the participants were separated in two rooms and interacted through Skype. Affective behaviour expressed by the participants was annotated with time- and value- continuous emotional dimensions (arousal and valence) by six French-speaking assistants, for the first five minutes of each recording, and for 46 participants. A web-based annotation tool, ANNEMO was used to perform emotion ratings and the annotations were done separately for each emotional dimension, using a slider with values ranging from -1 to +1 and a step of 0.01, obtained values were resampled at a frame rate of 40ms. In order to obtain a single gold-standard from the pool of ratings, each trace (i.e., a time- and value-continuous emotional annotation) was first assigned a weight according to the agreement of this rating with the five others. The final gold-standard was obtained by simply averaging the traces, after normalisation according to the inter-rater agreement.

To evaluate our model using Recola, we need to convert the continuous annotation of arousal and valence to discreet annotations including five level cf. Table 3. For that we divided the interval (between -1 and +1) of the continuous annotation to five levels as follows:

- level 1: annotation value between -1 and -0.6
- level 2: annotation value between -0.6 to -0.2
- level 3: annotation value between -0.2 to +0.2
- level 4: annotation value between +0.2 to +0.6
- level 5: annotation value between +0.6 to + 1

Table 4: Percentage of each annotation level for arousal and valence.

	Level 1	Level 2	Level 3	Level 4	Level 5
Arousal	0%	19%	67%	14%	0%
Valence	0%	2%	80%	18%	0%

4.3.2 Results

Figure 29 shows the training and validation loss and Figure 30 indicates the training and validation accuracy. Both loss and accuracy become steady after 20 epochs. For this, we train a new network from for 20 epochs and then evaluate it on the test set. Loss and accuracy of the model of arousal and valence prediction are given in Table 4.

[4] https://diuf.unifr.ch/diva/recola/index.html

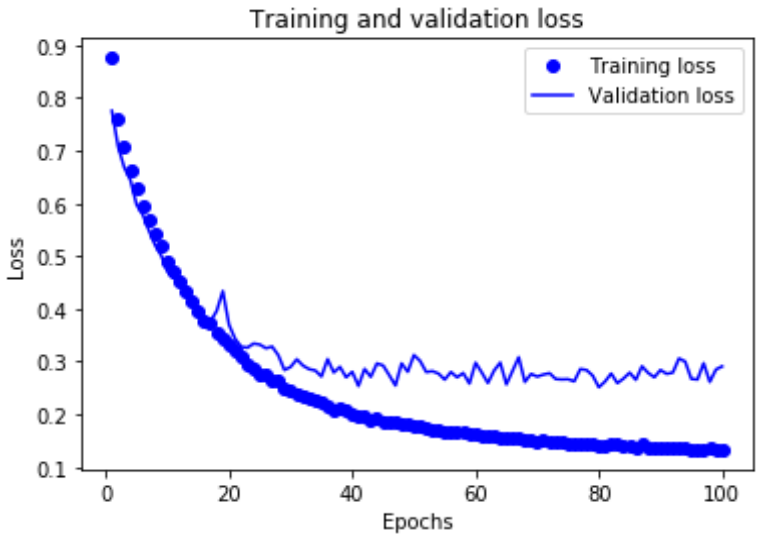


Figure 28: Training and validation loss.

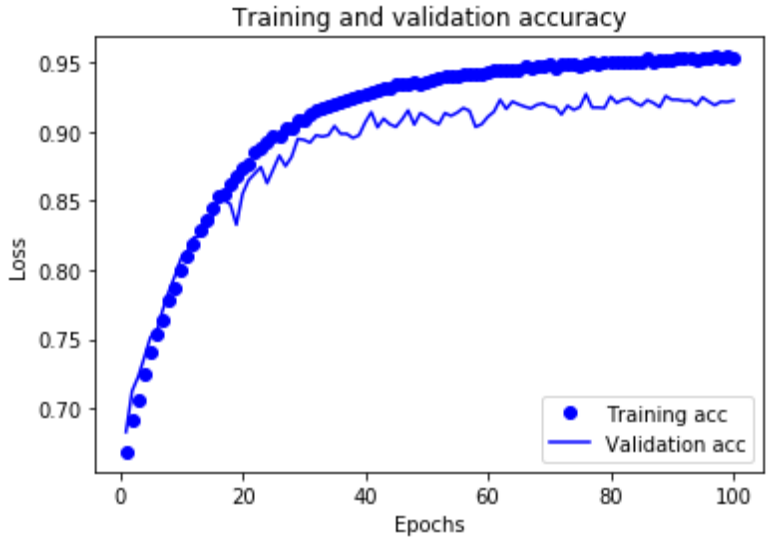


Figure 29: Training and validation accuracy.

Table 5: Loss and accuracy for arousal and valence.

	Loss	**Recall**
Arousal	0.87	0.66
Valence	0.71	0.65

4.4 Cognitive Behaviour

4.4.1 Experimental Setup

To train our model we make use of the NoXi database. NoXi (NOvice eXpert Interaction) database consists of multilingual natural dyadic interactions (Cafaro, 2017). It is publicly available through a web interface[5]. It provides spontaneous interactions that involve an expert and a novice discussing about a given topic (e.g. sports, politics, videogames, travels, music, etc.). The dataset contains over 25 hours of dyadic interactions spoken in multiple languages (mainly English, French, and German). The French part of NoXi database (which is composed of 21 sessions with a total duration equal to 7 hours and 25 minutes) was annotated using Poggi's definition of engagement. The engagement of both expert and novice was continuously annotated (Dermouche S., 2018). To facilitate the task of continuous annotation, we had defined five levels of engagement:

1. Strongly disengaged;
2. Partially disengaged;
3. Neutral;
4. Partially engaged;
5. Strongly engaged;

In order to avoid the biases of the verbal behaviour when annotating engagement, we had filtered it out, for both expert and novice by applying a Pass Hann Band Filter. Table 5 gives the percentage of each engagement level for expert and novice.

Table 6: Percentage of each engagement level in NoXi database.

	Level 1	Level 2	Level 3	Level 4	Level 5
Expert	1.25%	6.33%	14.32%	71.56%	6.51%
Novice	2.34%	10.12%	24.78%	58.05%	4.68%

4.4.2 Results

Loss and accuracy of our model to predict engagement level are given in Table 6.

Table 7: Loss and accuracy of engagement prediction for expert and novice.

	Loss	Recall
Expert	1.23	0.5
Novice	1	0.52

[5] https://noxi.aria-agent.eu

5 Discussion

5.1 Main Findings

In the previous sections we described the methods and the achieved results for the inference of short-term behaviours based on multimodal sensor data.

Regarding the short-term physical behaviour, we developed a model for counting steps and we managed to detect steps during a short-time period, from 1 minute to 60 minutes (one hour). However, we found that the best prediction score is achieved for counting steps during one hour or even more. Furthermore, we developed a model based on accelerometer data in order to detect every day activities during a short-time period (one minute). However, we found that the activities taking the bus, cycling, walking, being still (standing, sitting) and tilting can be better detected by using the Google API model for activity recognition, and thus, we decided to use this one for the COUCH system.

Regarding the social behaviour model, we used a customized model in order to detect if the subject interacts with others or not. This model calculates for every minute the social level of a subject using different types of data. We found that socially active interactions can be detected quite accurately, however, there is still a room for improvement for the inference of social isolation.

Regarding the emotional and cognitive behaviour model, we made use of pre-recorded sessions from the RECOLA and NoXi corpus for training the neural network model and predicting the arousal, valence values and the engagement. We found that these behaviours can be detected and predicted quite accurately, and as such propose to use these models in COUCH.

5.2 Open Issues

Concerning the short-term physical behaviour, we decided to use the Google API model. However, a drawback of the Google API model is the limited number of the activities that can be detected. Thus, we will try to enhance the classification performance of our own model for the activity recognition and detect more everyday activities, such as standing, lying, walking on stairs and running. A limitation of our activity recognition model is related to the acquired data, since we recruited only 8 subjects and we asked them to perform the activities for a few minutes. Consequently, in future work, we will analyse data from more subjects. Furthermore, we will try different techniques to process raw data, including different window segments and we will evaluate the performance of the prediction model, comparing the Random Forest classifier with other classification algorithms.

Concerning the short-term social behaviour, we used smartphone data (such as Bluetooth, ambient noise, GPS, number of calls and number of SMS) in order to detect the social activity of a user every minute. However, our customised model performs poorly for predicting whether a user is social inactive. For this reason, we will try to enhance the model's performance in future work, and consider more types of data, such as Wi-Fi signal, battery level and more patterns of using a smartphone device.

Concerning the detection of emotional and cognitive behaviours, since there was a lack of real-time data, we made use of pre-recorded sessions from the corpus. In the future we will try to verify our model in detecting and predicting these behaviours in real-time.

6 Bibliography

Activity Recognition API. (2018). Retrieved from Google: https://developers.google.com/location-context/activity-recognition/

Baltrušaitis T., M. M. (2015). Cross-dataset learning and person-specific normalisation for automatic Action Unit detection. *IEEE International Conference on Automatic Face and Facial Expression Recognition and Analysis Challenge, , IEEE International Conference on Automatic Face and Gesture Recognition*.

Cafaro, A. W. (2017). The NoXi database: multimodal recordings of mediated novice-expert interactions. *19th ACM International Conference on Multimodal interaction*.

Camurri, A. C. (2004). Toward real-time multimodal processing: EyesWeb 4.0. *Proceedings of the artificial intelligence and the simulation of behaviour (AISB)*.

Corrigan, L. J. (2016). Engagement perception and generation for social robots and virtual agents. *Toward Robotic Socially Believable Behaving Systems*, 29-51.

Dermouche S., ,. P. (2018). From analysis to modeling of engagement as sequences of multimodal behaviors. *LREC*.

Ekman., P. (1997). What we have learned by measuring facial behavior.,. In *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)* (pp. 469–485).

F. Attal, S. M. (2015). "Physical Human Activity Recognition Using Wearable Sensors," Sensors, vol. 15, no. 12, p. 29858.

Gerwin Huizing, K. K. (2018). *D7.1: System architecture and design of APIs*. The Council of Coaches Consortium.

Glas, N. a. (2015). Definitions of engagement in human-agent interaction. *International Conference on Affective Computing and Intelligent Interaction*, (pp. 944–949).

Kroschel., M. G. (2005). Evaluation of natural emotions using self assessment manikins. *IEEE Workshop on In Automatic Speech Recognition and Understanding* (pp. 381–385). IEEE.

Labrador, O. D. (2013). A Survey on Human Activity Recognition Using Wearable Sensors. pp. 1192-1209.

M. Gjoreski, H. G. (2016). "How Accurately Can Your Wrist Device Recognize Daily Activities and Detect Falls?," Sensors (Basel), vol. 16, no. 6.

M. Shoaib, S. B. (2015). "A Survey of Online Activity Recognition Using Mobile Phones," Sensors, vol. 15, no. 1, p. 2059.

Oresti Banos, K. K. (2018). *D4.1: State-of-the-art, requirement analysis and initial specification of the Holistic Behaviour Analysis Framework*. The Council of Coaches Consortium.

P. Ekman, W. V. (2002). *Facial action coding system*. Salt Lake City.

Pandas. (2018). *Pandas: powerful Python data analysis toolkit*. Retrieved from pandas.pydata.org: https://pandas.pydata.org/pandas-docs/stable/

Peters, C. P. (2005). Engagement Capabilities for ECAs. *AAMAS'05 workshop Creating Bonds with ECAs*.

Poggi, I. (2007). *Mind, Hands, Face and Body: A Goal and Belief View of Multimodal Communication*.

Ringeval F., S. A. (2013). Introducing the RECOLA Multimodal Corpus of Remote Collaborative and Affective Interactions. *2nd International Workshop on Emotion Representation, Analysis and Synthesis in Continuous Time and Space (EmoSPACE 2013)*. IEEE.

scikit-learn. (2018). *1.13. Feature selection*. Retrieved from http://scikit-learn.org/stable/modules/feature_selection.html

scikit-learn. (2018). *scikit-learn: Machine Learning in Python*. Retrieved from www.scikit-learn.org

scipy.stats.pearsonr. (2018). Retrieved from https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.pearsonr.html

Sidner, C. L. (2005). A first experiment in engagement for human-robot interaction in hosting activities. *Advances in natural multimodal dialogue systems*, 55-76.

Sidner, C. L. (2005). Explorations in engagement for humans and robots. *Artificial Intelligence*, 166(1-2):140–164.

Stackoverflow. (2018). *Scikit-learn: How to obtain True Positive, True Negative, False Positive and False Negative*. Retrieved from Scikit-learn: https://stackoverflow.com/questions/31324218/ scikit-learn-how-to-obtain-true-positive-true-negative-false-positive-and-fal

Tapia, E. M. (2008). "Using machine learning for real-time activity recognition and estimation of energy expenditure," Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts and Sciences., Massachusetts Institute of Technology.

Y. P. Chen, J. Y. (2008). "Onlineclassifierconstructionalgorithmfor human activity detection using a tri-axial accelerometer," Applied Mathematics and Computation, vol. 205, no. 2, pp. 849-860.

Yu, L. L. (2016). Building Chinese Affective Resources in Valence-Arousal Dimensions. . *HLT-NAACL*.

Zeng, Z., & Pantic, M. R. (2009). A survey of affect recognition methods: Audio, visual, and spontaneous expressions. . *IEEE transactions on pattern analysis and machine intelligence*, 31(1), pp.39-58.