Managing Affective-learning THrough Intelligent atoms and Smart InteractionS

D4.5 Multimodal learning analytics (M21)

Workpackage	WP4 - Affective and Natural Interaction Instruments
Editor(s):	Mohammad TAHERI, NTU Caroline LANGENSIEPEN, NTU Enrique HORTAL, UM Esam GHALEB, UM Dorothea TSATSOU, CERTH Nadia POLITOU, ATOS Ana Luiza PONTUAL, ATOS Miquel MILA, ATOS
Responsible Partner:	NTU
Quality Reviewers	DXT, OTE
Status-Version:	Final – v1.0
Date:	Project Start Date: 01/01/2016; Duration: 36 months Deliverable Due Date: 30/09/2017 Submission Date: 14/11/2017
EC Distribution:	Public
Abstract:	This document presents the first version of the MaTHiSiS learning analytics. Affect cues are inputs into a comprehensive multi-modal fusion algorithm which uses a multi-layered approach and comprehensive genetic algorithm to classify leaner affect state in a summative view of “Bored”, “Frustrated” or “Engaged” with a corresponding confidence rate. Learner progression through the learning material is displayed through learning analytics. Summative information

This document is issued within the frame and for the purpose of the *MATHISIS* project. This project has received funding from the European Union’s Horizon 2020 Programme (H2020-ICT-2015) under Grant Agreement No. 687772

	from the learner profile of the previous engagements of the learner with similar material, are visualized in graphs in the backend of MaTHiSiS. The visualization of learning analytics allows teachers and carers to monitor their learners, through different learning materials and in different classrooms.
Keywords:	Sensorial data, Multimodal fusion, Learning analytics
Related Deliverable(s)	Dependent on work in "D4.1 MaTHiSiS Sensorial Component" and "D4.3 Affect Understanding in MaTHiSiS"

Document History

Version	Date	Change editors	Changes
0.1	26/07/2017	Mohammad Taheri (NTU)	Initial ToC.
0.2	31/07/2017	Mohammad Taheri (NTU)	Updated ToC after meeting.
0.3	03/08/2017	Mohammad Taheri, Caroline Langensiepen (NTU)	NTU data flow section and peripheral interaction
0.4	10/08/2017	Mohammad Taheri (NTU)	NTU performance section
0.5	17/08/2017	Mohammad Taheri (NTU)	Renaming NTU sections in document
0.6	07/09/2017	Mohammad Taheri (NTU)	Updating as per comments
0.7	10/09/2017	Enrique Hortal, Esam Ghaleb (UM)	Multi-modal fusion
0.8	11/09/2017	Dorothea Tsatsou (CERTH)	Section 3.1 – Sensorial data
0.9	13/10/2017	Nadia Politou, Ana Luiza Pontual, Miquel Mila (ATOS)	Contributions to section 5, Learning Analytics
0.9.1	20/10/2017	Mohammad Taheri (NTU)	Collation and integrity
0.9.2	27/10/2017	Mohammad Taheri (NTU)	Corrections after internal review
0.9.3	01/11/2017	Mohammad Taheri (NTU), Esam Ghaleb (UM), Dorothea Tsatsou (CERTH), Nadia Politou (ATOS)	Partner corrections and consolidation
0.9.4	09/11/2017	Mohammad Taheri	Corrections after second round of internal review
0.9.5	11/11/2017	Nelly Leligou (OTE)	New version after third round of internal review
0.9.6	14/11/2017	Ana Piñuela (ATOS)	Final quality review

Version	Date	Change editors	Changes
1.0	14/11/2017		FINAL VERSION TO BE SUBMITTED

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
ASLAW	Average Smart Learning Atom Weight
IPA	Interaction with Platform Agents
ITC	Industrial training case
JSON	JavaScript Object Notation
LA	Learning Action
LG	Learning Graph
MEC	Mainstream education case
MM	Multi-modal
PA	Platform Agent
PMLDC	Profound and multiple learning disabilities case
SC	Sensorial Component
SLA	Smart Learning Atom
SVM	Support Vector Machine
UM	Maastricht University
xAPI	Experience API

Table 1 Definitions, Acronyms and Abbreviations

Project Description

The MaTHiSiS learning vision is to provide a novel advanced digital ecosystem for vocational training, and special needs and mainstream education for both individuals with an intellectual disability (ID) and neuro-typical learners. This ecosystem consists of an integrated platform, along with a set of re-usable learning components with capabilities for: i) adaptive learning, ii) automatic feedback, iii) automatic assessment of learners' progress and behavioural state, iv) affective learning, and v) game-based learning.

In addition to a learning ecosystem capable of responding to a learner's affective state, the MaTHiSiS project will introduce a novel approach to structuring the learning goals for each learner. Learning graphs act as a novel educational structural tool. The building materials of these graphs are drawn from a set of Smart Learning Atoms (SLAs) and a set of specific learning goals which will constitute the vertices of these graphs, while relations between SLAs and learning goals constitute the graph's edges. SLAs are atomic and complete pieces of knowledge which can be learned and assessed in a single, short-term iteration, targeting certain problems. More than one SLA, working together on the same graph, will enable individuals to reach their learning and training goals. Learning goals and SLAs will be scoped in collaboration with learners themselves, teachers and trainers in formal and non-formal education contexts (general education, vocational training, lifelong training and specific skills learning).

MaTHiSiS is a 36 month long project co-funded by the European Commission Horizon 2020 Programme (H2020-ICT-2015), under Grant Agreement No. 687772

Executive Summary

D4.5 describes work completed as fulfilment to the first iteration of Task 4.3 “Multimodal learning analytics” and is intended to explain to the technical reader a) how multi-modal fusion towards affect state detection is implemented as well as to describe the operation of the learning analytics module of the MaTHiSiS system. Task 4.3 is closely related to the work done in “Task 4.1 MaTHiSiS sensorial component” and “Task 4.2 Affect understanding through interaction with learning material”; it makes direct use of the data outputs of both these tasks which focus on affect state detection and attribute extraction based on individual modalities. Multi-modal fusion is expected to improve the performance of the affect detection (compared with individual modalities). Our aim is to identify multi-modal techniques and optimise them per use case towards improved performance.

Here, in Task 4.3, sensor data (facial expressions, eye gaze, body pose, voice) from task 4.1 and specific interaction and peripheral behaviour data from task 4.2 (leaner’s response to questions / quiz challenges and response times) are fused in a comprehensive multi-modal fusion algorithm where the indicators of leaner’s engagement are extracted using computational intelligence algorithms. Subsequently, the engagement data is used to develop complex features that carry inferences to the leaner’s emotional affect state as an outcome. The affect detection and tracking methodology which is currently integrated into the system is a key component of MaTHiSiS’ learner engagement tracking component. Affect cues are inputs into a comprehensive multi-modal fusion algorithm which uses a multi-layered approach and comprehensive genetic algorithm to classify leaner affect state in a summative view of the ongoing emotional state of the learner which is presented as either “Bored”, “Frustrated” or “Engaged” with a corresponding confidence rate.

The learner’s interaction with the learning material is monitored, and correct and incorrect responses are logged, resulting in a live ‘score’ value, which represents the learner’s performance. Using robust analysis not only the learner performance in terms of a score (which is currently provided by other learning management systems) can be shown; as surplus, the learners’ engagement with the learning material through different facets like learning progress and effort are displayed in a) easy to view graphs b) with various levels of time granularity and c) classroom or learner ID filters for the teacher or carers to review individual and classroom progress at any live moment in time and in the previous learning session.

In conclusion, task 4.3 aims to deliver learning analytics through complex engagement tracking through the lens of sensory data and leaner’s responses to the learning material. The first version of the work done in Task 4.3 has been described in this document, a second revision will be fulfilled in D4.6 MaTHiSiS Learning Analytics. It is important to note that the current implementations will be validated and evaluated in the assisted pilots phase and the collected feedback will be used to refine the current designs and provide a highly valuable (next) version of these components. The first publications of the innovative multi-modal fusion techniques defined in MaTHiSiS have already been submitted at the time of writing the deliverable.

1. Introduction

1.1 Context and scope

Task 4.3 (Multi-modal learning analytics) follows closely and in interconnection with other WP4 tasks T4.1 and T4.2. In this document the WP4 dataflow is described, and the data flow relationships between the different tasks in WP4 is explained in Figure 1 and Figure 2. The process of fusing the sensory data to classify learner affect state is described. Learning performance is defined, and complex examples of learning analytics are explored through the MaTHiSiS system.

In task 4.3 we focus a) on learning analytics and b) on multi-modal fusion towards affect state detection. The **important contributions** of this deliverable which reports the efforts in task 4.3 are:

- The description of the learning analytics that will be presented in MaTHiSiS platform mainly for the tutor; the added value here is that instead of showing just the scores of the tests undertaken by the learner, in MaTHiSiS we provide much richer information considering affect state apart from scores. Example of MaTHiSiS specific metrics are:
    - The learner performance in terms of weights of the learning graphs which reflect the average progress of the learner in the learning graph.
    - The learner performance in terms of time required to accomplish a learning experience which reflects the quality of individual achievements in a learning graph.
- The description of multi-modal fusion component; this receives input from the individual modalities dealt with in task 4.1 and employing state-of-the-art feature detection algorithms it detects the affect state of the learner with enhanced (compared to individual modalities) performance.

The components designed and implemented in this task include live graphs with varied levels of granularity ‘hour’, ‘day’ and ‘month’ which present the results of the learner’s progress and performance in the learning experience to the tutors to support their decision making. The correlation of task 4.3 and the relevant data flow in Work package 4 “Affective and Natural Interaction Instruments” is shown in Figure 1.

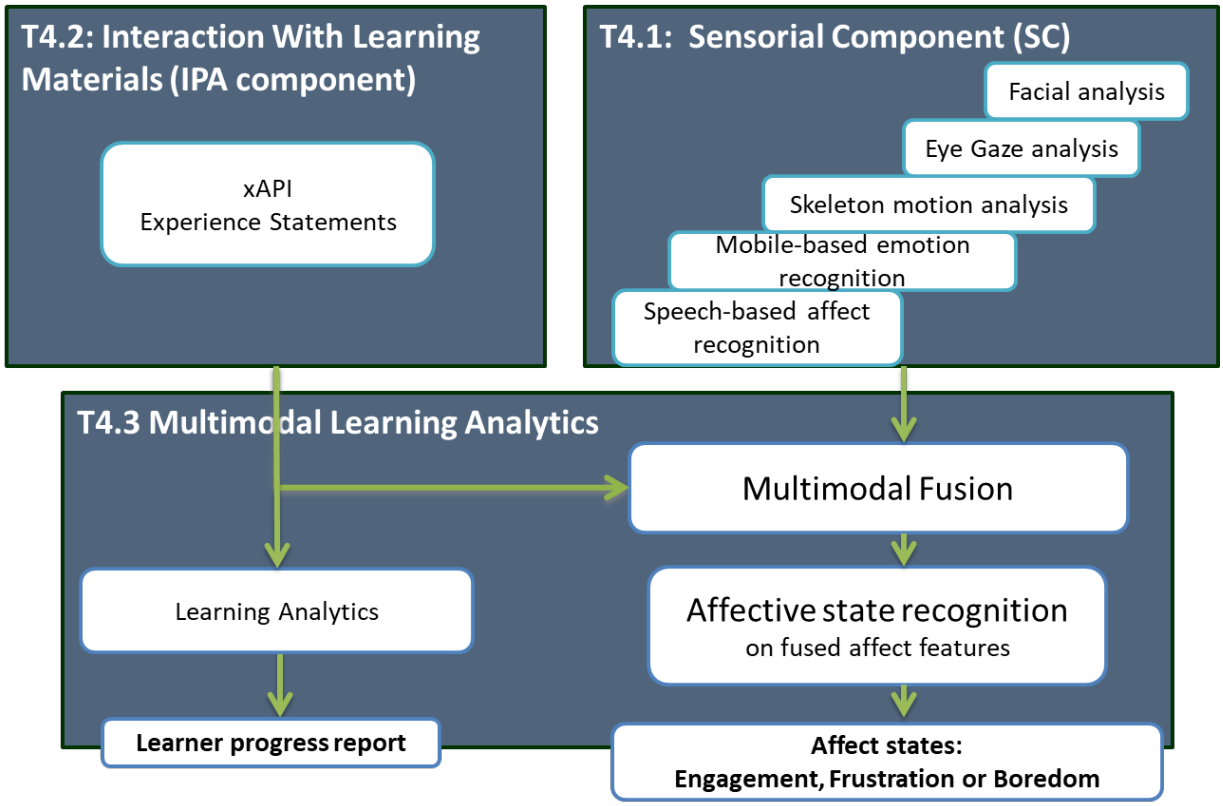

Figure 1: Work package 4 relationships between tasks

Learner's multi-modal sensory data is collected in T4.1, this data models the learner's engagement in the learning material. T4.2 collects the learner's responses to questions and quizzes with correct, incorrect or passes being logged along with response times. The learner's responses to the learning material are stored in the widely adopted 'xAPI' standard. In tasks 4.3, these xAPI statements represent the learning performance in respect to the learning material and are later used to display comprehensive charts and graphs in the Learning Analytics component. Additionally, T4.3 combines the learner's emotional state (as detected through individual modalities) with the learner's performance to model the learner and identify where they are Bored, Engaged or Frustrated. A more detailed description of the dataflow between all tasks of WP4 is provided in the sequel in section 1.3.

It is stressed that:

- Documentation and a user-friendly manual through the learning analytics front-end are included in the resources provided to the pilot users which are 'living resources' in the sense that MaTHiSiS consortium continuously updates them to the extent possible responding to the pilots' needs.
- The components designed and implemented up to now will be evaluated during the assisted pilots by real users. Based on this feedback as well as comments received from the consortium pilot partners, potential improvements will be introduced in the next version of the platform and described in the next version of the deliverable.
- MaTHiSiS attempts affect detection in the wild and for this reason multi-modal fusion is pursued to increase the performance of the MaTHiSiS affect detection component. The datasets used to train the algorithms in this first phase were datasets publicly available because the individual sensorial input data collected in MaTHiSiS from some use cases did not allow us to study or train the models due to the resulted dominance of one specific label which is Engagement. We anticipate that after the assisted pilots, we will have collected valuable data to train the multi-modal algorithms.

1.2 Structure

The current deliverable is organised in chapters as follows:

- Chapter 2 presents the datasets created from the retrieved signals used to train the multi-modal classification component.
- Chapter 3 provides a literature review of the state-of-art in multi-modal fusion methods and algorithms, describes the multi-modal fusion approach and machine learning methods adopted and already implemented in the MaTHiSiS platform to fuse the retrieved signals.
- Chapter 4 introduces the metrics and the approaches adopted to assess the learner performance as MaTHiSiS is putting the learner at the center to track their affective state and overall performance.
- Chapter 5 includes the technical description of the learning analytics component and examples of its output to provide a more concrete description of how MaTHiSiS displays learner performance to the learning supervisors; teachers, instructors etc.
- Chapter 6 provides conclusions.

1.3 Dataflow description

This following schematic gives a brief overview of the existing data channels both input and output between T4.2 and T4.3 in more detail. A summary of what happens to the data in every step is the main purpose of this section. In T4.3 affect data, bearing learner's affect state is fused with interaction data to provide insight into the learner's engagement with the learning material, this is then stored in the Learner Profile.

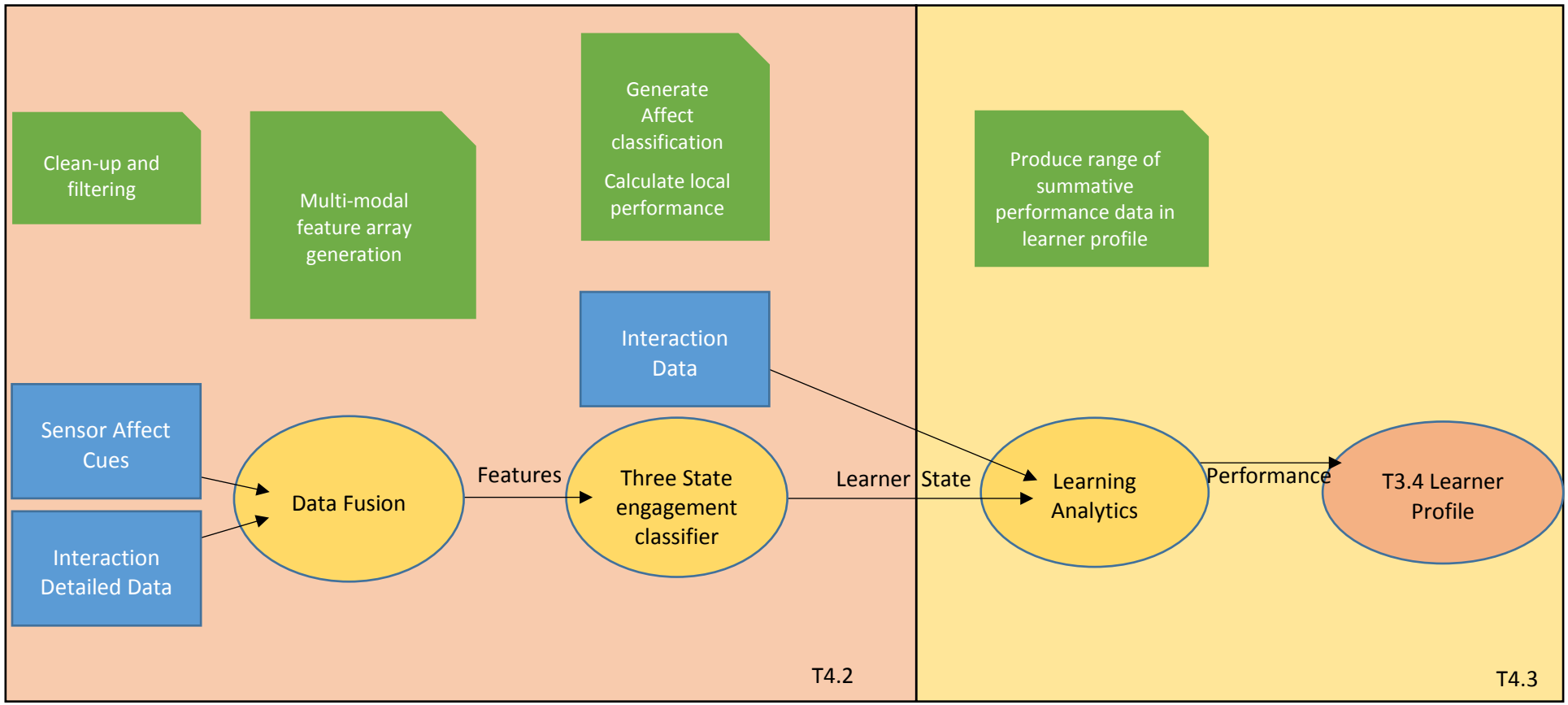

Figure 2 Data flow overview of relationship between T4.2 and T4.3

T4.1 provides a base for the affect understating by recognizing affect features as distilled from the sensorial information which has provided cognitive responses from the direct interaction with the current learning material. Affect cues are inputs for T4.2 into a comprehensive multi-modal fusion algorithm which combines these features into a single stream of information ready for classification. A multi-layered approach uses a comprehensive genetic algorithm to classify leaner affect state in a summative view of “Bored”, “Frustrated” or “Engaged” with a corresponding confidence rating. This multi-modal approach provides the fundamental input to the multi-modal fusion of affect states and the classification of learner engagement level in this task.

T4.2 provides this real-time information about the learner interacting with the learning material. The learner’s interaction with the learning material is monitored and correct and incorrect responses are logged, resulting in a score, which represents the learner’s progress. This information is then collected, and comprehensive performance analytics deliver a performance score. The learner engagement and learner performance information from their response time and response challenge are all inputs into the learning analytics. Performance calculation of learning activities has been described in detail in this document, which later forms the basis of the learning analytics in combination with learner affect state.

Learning analytics is where summative information from the learner profile of the previous engagements of the learner with similar material, at similar levels of difficulty, and the outcomes of previous interventions are analysed. This historic data is collected and analysed in the backend of MaTHiSiS to be afterwards visualized in graphs in a dashboard. The visualization of learning analytics allows teachers and carers a) to monitor their learners b) to compare them in local and global level with other students, c) to compare their performance in learning materials and classrooms. Learner analytics will provide the metrics of the learner engagement in a more detail of subject, classroom and school scopes. This provides individual leaner and classroom based information for tutors. Examples of these learning analytic displayed have been included in the contents of Chapter 5.

2. Dataset description

This section describes the existing data that contribute to T4.3, as well as the sources of this data, and enumerates the specifications of these inputs.

2.1 Sensorial data

This section describes the features and labelled data obtained from the Sensorial Component, work done in T4.1, and outlined here for reference as to what pertains the input of T4.3 in terms of sensorial data. This section will aggregate all relevant information from deliverables D4.1 and D4.2 (MaTHiSiS sensorial component M12 and M18 respectively).

2.1.1 Data description

All five modalities that capture affective data from agents' sensors predict affect labels in the three **affective states** of the Theory of Flow (namely **engagement, frustration, boredom**), through mechanisms described in deliverable D4.2, ch. 3, used by the algorithms developed in T4.3 and detailed in subsequent sections of this deliverable, for late fusion. Each modality also outputs a set of affect-related features, which drives the affect label prediction and may also be used per se for early multimodal fusion. These features are described in Table 2, comprising an update (where applicable) of D4.1's section 3.1.

Modality	Affect related features	Description
Facial expressions analysis **Graph-based method**	1x98 vector/frame.	Eigen vectors of first and second largest for a specific frame, based on the Eigen decomposition of facial landmarks, organized in a connected graph. More details can be found in Section 2.1.2.1 of deliverableD4.1
Facial expressions analysis **Appearance -based method**	1X2112 vector/video.	The number of frames in video sequence can vary from 1, up to 500 frames. Fisher vector dimensionality is 2Kd which depends on the number of the GMM components (K), and the dimensionality of the used set of features. Then, the Fisher vector ϕ is computed by stacking the differences (the assignment of the local features to the first and second differences of GMM centres): $\phi = [\Phi(1)^{(1)}, \Phi(1)^{(2)}, ..., \Phi(K)^{(1)}, \Phi(K)^{(2)}]$. The value used for K = 16, and d is the length of SIFT histogram reduced by PCA from 128 to 64, augmented by the spatial information = 64+2=66. As a result, this length is: 2*66*16 = 2112
Gaze Estimation	1x3vector/frame	X, Y, Z coordinates in space of the subject's gaze direction. More details can be found in Section 3.2.2 of deliverable D4.2.
Mobile Device-	249 descriptors of 3D motion	3D motion: • Acceleration values and acceleration's derivative (Jerk) (x, y

Modality	Affect related features	Description
based emotion recognition	and 2D surface gestures	and z projections) • Spectrum analysis • 2D surface gestures: • Touch parameters • Stroke levels
Skeleton Motion Analysis	A 1x1400featurevector (histogram) /frame sequence	Currently, each frame sequence is defined as 60 frames per sequence. Histogram represents the temporal evolution of key postures captured within the sequence, based on skeleton actions. More details can be found in Section 3.4 of deliverable D4.2.
Speech recognition and speech-based affect recognition	1x34 vector/audio segment.	Vector contains features: Zero Crossing Rate, Energy, Entropy of Energy, Spectral Centroid, Spectral Spread, and Spectral Entropy, Spectral Flux, and Spectral Roll off, MFCCs, Chroma Vector and Chroma Deviation, as described in Section 2.5.2 of Deliverable D4.1.

Table 2 Affect related features per modality

2.1.2 Status of collected data

Prediction of affect state labels and extraction of affect-related features occurs on-the-fly per certain time intervals, usually spanning few seconds, as per the requirements of the multimodal fusion algorithms, through the different modalities incorporated in the Sensorial Component, as described in Deliverables D4.1 and D4.2. However, for training the aforementioned classifiers, public and MaTHiSiS-specific datasets have been collected. The data collection task organized within MaTHiSiS, as described in deliverables D4.1 and D4.2, has yielded an initial dataset, as described in deliverable D4.2, section 2. Indicatively, the summation of collected data is outlined in Table 3.

Use Case	Learners	Sessions	Annotated Sessions (/w affect labels annotation)
MEC	27	86	72
ASC	13	34	27
PMLDC	7	11	5
ITC	7	14	14
CGDLC	5	15	15
SUM	**59**	**157**	**133**

Table 3 MaTHiSiS dataset

Data collection will continue throughout the MaTHiSiS assisted pilots, to enrich current datasets and better populate the scarce mobile-based inertia sensors dataset and PMLDC use case.

2.2 Labelled interaction data features

The multimodal fusion mechanism, apart from the sensorial components described in section 2.1, it takes into account the Interaction with Platform Agents (IPA) component. The feature extraction, modelling and classification based on the Experience API (xAPI), is extensively described in deliverable D4.3 [25]. The output of IPA represents the affect understanding through interactions with the PA while performing a learning activity. The information provided for this component is used in the multimodal fusion as an input to infer the affective state of the learners. The IPA behaviour is widely described in [24] and [25]. The format elected to communicate the interactions in MaTHiSiS context is the Experience API or xAPI. This specification for learning technology was designed to collect data regarding the learner's experience while interacting with learning content. The component is widely described in Annex I of the deliverable [25].

2.2.1 Labelled peripheral input data features

Peripheral data collected will be from mainly two sources

- Pointer location on screen
- Keyboard inputs

Peripheral data features are calculated using data from the start of leaner being asked a question until the learner responds or the question response time, time-outs. 5 seconds of data is used in the feature calculation and the calculation is done as frequently as required by the fusion algorithms.

Pointer features:

- Impulsivity: The quick succession of presses in close proximity of time and space.
- Definiteness: For a single press the speed of the press time calculated from the start of challenge.
- Line discrepancy: Average divergence in path from the shortest path, the line formed from the first click and last click in a challenge. It is assumed all clicks are intentional and needed.
- Dwell: The closeness in proximity of pointer locations for an extended length of time.
- Absence of movement: Length of time with no movement in the pointer.

Keyboard:

- Impulsivity: The quick succession of key presses in close proximity of time.
- Definiteness: For a single press the speed of the press time calculated from the start of challenge.

3. Multimodal fusion

3.1 State of the art

The recent technological advancements brought interactivity between people and digital devices to a completely different level, making computers and mobile phones an important part of our daily lives. A natural way in which people communicate with each other is based on emotions. Therefore, there is an increased interest in the human computer interaction (HCI) field towards enhancing digital devices with emotion recognition abilities for obtaining a more natural HCI experience. Emotions can be expressed using both verbal and non-verbal cues such as facial expressions, gestures or the tone of the voice. Facial expressions represent one of the most significant cues for recognizing emotions, due to their universality proven by Ekman [1] who found that six basic emotions (happiness, fear, sadness, disgust, surprise and anger) are the same across cultures. The applications of an automatic facial recognition system go beyond HCI, being useful also in website customization, gaming industry, humanoid robots, as well as in improving online education systems. Due to the potential applications of such a system, many research studies have been done in classifying faces in still images [2] or in video sequences [3] into one of the six basic emotions.

Data mining algorithms employed for recognizing the six basic emotions have been rather successful on posed datasets gathered in controlled environments such as the Cohn-Kanade [4], the JAFFE [5], the CMU Pose Illumination and Expression (PIE) [6] or the MMI database [7]. While recently, efforts were devoted to more challenging datasets, captured in uncontrolled spontaneous conditions such as the Acted Faces in the Wild (AEFW) dataset [8], containing video clips of unconstrained facial expressions, with varied head poses, occlusions and challenging illumination conditions. The palette of feature extraction techniques employed for facial expressions recognition contains appearance based methods (Gabor filters [9], LBP [10], SIFT [11]), geometric features [12] and also unsupervised feature learning methods such as the recently adapted CNN models [13]. On top of the extracted features, multiple classification algorithms are used, varying from SVM with different kernel methods [14], neural networks [3], Boltzaman machines [15] to deep architectures [16]. Apart from the visual modality, audio-based emotion recognition is also promising and features such as prosody, jitter, or the fundamental frequency proved to be useful [17].

Furthermore, studies in multi-modal emotion recognition showed the benefits of fusing visual and acoustic information [18], due to the complementarity of the two modalities. Therefore, in the context of task 4.3 we have implemented and analysed a multi-modal framework for emotion recognition from video sequences, by taking advantage of both visual and audio features. Moreover, one of the main contributions of multi-modal fusion consists of proposing a hierarchical fusion approach, which combines feature level and decision level fusion in an efficient manner, using information gain principles, which is depicted in Figure 3. The proposed fusion framework is generally enough to be useful also for other tasks such as behaviour or object recognition, as long as there are available different types of features which are complementary. In MaTHiSiS context, this method will be extended for the fusion of different SC modalities and will be adapted to the particular traits of each use case.

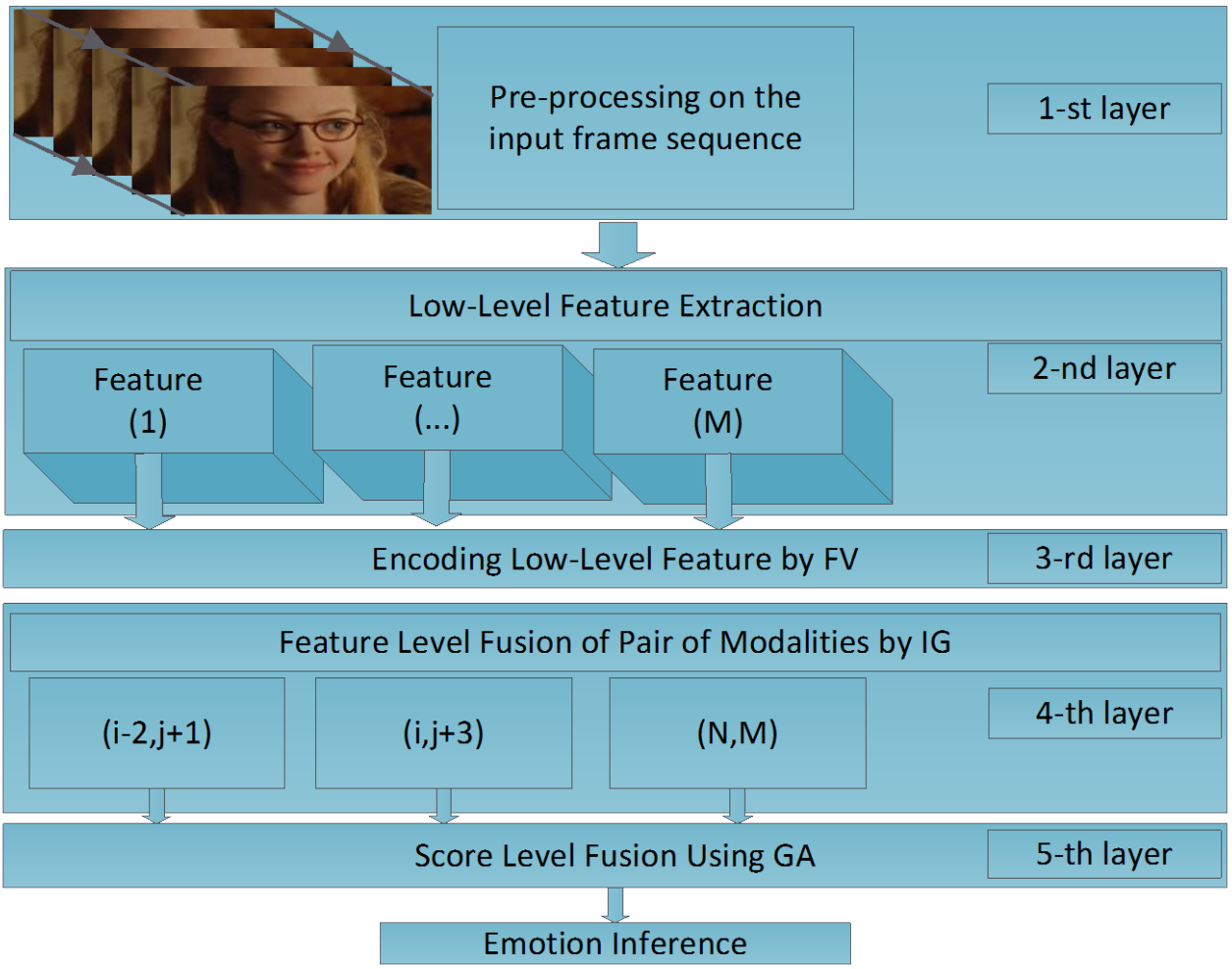

Figure 3 Hierarchical multimodal fusion framework based on feature level and score level fusion.

The proposed scheme starts with a pre-processing layer for face and facial landmark detection, and face alignment. This layer is followed by extracting a set of M low-level features (e.g. DSIFT, CNN, and geometric features). The third and fourth layers include high level representation of features by Fisher Vector encoding (FV) and selecting pair of modalities based on information gain principles (IG). In the fourth layer are included examples of the selected pair modalities having indices (i; j) 2 M. The last layer of the framework depicts score level fusion optimized using a Genetic Algorithm (GA).

In this approach, we take advantage of different feature extraction algorithms, extracted from the audio channel and also from the entire face or from salient facial regions of interest (ROI), (e.g. eyes, nose, mouth, forehead and chin), such as dense scale invariant feature transformation (DSIFT), geometric features, and a pre-trained CNN model for face recognition provided by the Visual Geometry Group (VGG-face) [16], denoted as a set of M features on the 2-nd layer of Figure 3. Each of these features are useful, while one constraint in fusing them is given by their different underlying probabilities and ranges. Another contribution of this paper refers to encoding the different features using Fisher Vector [19] representations, which are useful at projecting all types of features in the same space and also as it facilitates the analysis of videos with different lengths, while efficiently capturing the facial dynamics (the 3-rd layer in Figure 3). Next, we use an efficient algorithm for feature-level fusion, which finds the best types of features to be fused in a hierarchical manner, based on minimizing the Kullback-Leibler (KL) divergence [20] between the probability distribution function (PDF) of true labels and the PDF of predicted labels, obtained after employing a classification algorithm. For example, at a first stage the mouth region features and audio features are fused in a new feature vector and also DSIFT features and geometric features are fused in another one. Then, at the next stage (the 5-th layer in Figure 3), the two new obtained feature

vectors are fused using a decision-level fusion algorithm which optimizes the weights of each modality using a Genetic Algorithm (GA).

The proposed framework is useful, as, instead of fusing all features at an early stage as described by [21] or at the end of the pipeline as proposed by [22], it searches for the best combinations at different processing stages for finding complementary modalities. Furthermore, the use of a genetic algorithm facilitates finding the optimum weights for the decision level fusion. We evaluated our proposed approach on the challenging AEFW dataset [23] and compared it with a deep learning architecture.

3.2 Multimodal Fusion Framework for Affect Detection

As an initial framework, for multimodal fusion within MaTHiSiS, we have conducted extensive study using publicly available dataset due to the lack of available dataset (at that time) within MaTHiSiS project. In this chapter, we firstly explain the pre-processing phase of facial images and how we obtain a face track from a video. Then, we present the low-level feature extraction methods implemented in our framework for different modalities: audio and visual (geometric and appearance features). Finally, we describe the feature encoding and representation by means of Fisher vectors for video modelling and projecting features into the same space. These input channels serve as different modalities, where the aim is to fuse them for enhanced emotion recognition whether in later or early fusion. This chapter provides the two methods we studied, and outline their advantages and challenges. In addition, we aim to clearly show the powerful aspect of both schemes. Further details about multimodal fusion within MaTHiSiS are provided in section 3.4.

3.2.1 **Pre-processing**

Facial Landmark Detection: Succeeding the step of face detection, we detect 49 landmarks and track them in each frame of a video, using the Supervised Descent Method (SDM) [26]. SDM is a successful face shape regression technique, which begins with an initial $\mathbf{S}_0$ face shape and progressively predicts the final shape of the facial landmarks in an iterative way. Comparing to other techniques, this method provides robust and accurate landmark positions in challenging conditions, such as varying illumination and pose, and low quality images. In addition, it gives a reliable and robust tracking of facial landmarks in the wild, in real-time.

Face Alignment: Face alignment is an essential step in facial emotion recognition. It is the process of registering faces with respect to facial landmarks (e.g. eyes, nose, mouth, and chin) of the canonical frame. This process fixes the landmark positions in aligned images and it is carried out by similarity transformation. In our work, we use facial landmarks provided by SDM landmark detector and perform a similarity transformation that aligns faces to the fixed canonical frame based on eye centres positions. In addition, facial images are cropped and re-sized to a fixed resolution: 224 × 224. Examples of aligned and cropped faces are depicted in Figure 4, while Figure 5 presents examples of tracked facial images from the AFEW dataset.

Figure 4 Cropped and aligned faces from the AFEW dataset.

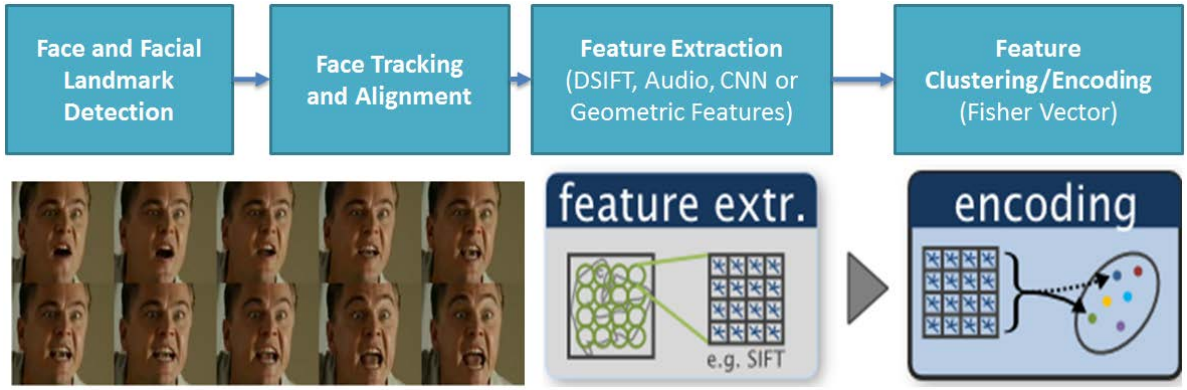

Figure 5 Face pre-processing, feature extraction and encoding.

3.2.2 Low-Level Feature Extraction

Emotion recognition relies on representative data along with accurate and discriminative descriptors. This type of information contributes in enhanced recognition and classification accuracy. Accordingly, this method extracts a set of low-level descriptors for the visual and audio modalities. Then, we use Fisher vectors for video modelling and projecting them into the same space. The low-level features used in our work are DSIFT, handcrafted geometric, CNN and audio features.

1) **DSIFT Features:** Dense Scale Invariant Feature Trans- form (DSIFT) has been widely used for image representation in the last decade, in many computer vision recognition tasks [27], [28]. In DSIFT, instead of sparsely detecting and selecting the facial key-points, we compute the DSIFT histogram densely over a given image, using a certain scale factor and step size. This has an advantage since it does not rely on facial landmark detection. We divide the facial images into a grid of overlapping blocks with a step size equal to 1. Specifically, the block size is 24 × 24. Later, we compute a DSIFT histogram for each block. This step is repeated in 5 scales, with a scale factor equal to √2.

In this work, we compute DSIFT with two approaches: (i) DSIFT on the entire facial image; and (ii) DSIFT on six distinct facial regions of interest (ROI): left eye, right eye, forehead, mouth, nose and the region between eyes. These six facial ROI are illustrated in Figure 6. We extract and crop the ROI using the facial landmarks provided by [26]. Then, DSIFT features are extracted from each region separately. In the reminder of this section, we refer to the DSIFT extracted from the entire facial image as DSIFT, while we call the DSIFT computed on ROI as ROI DSIFT.

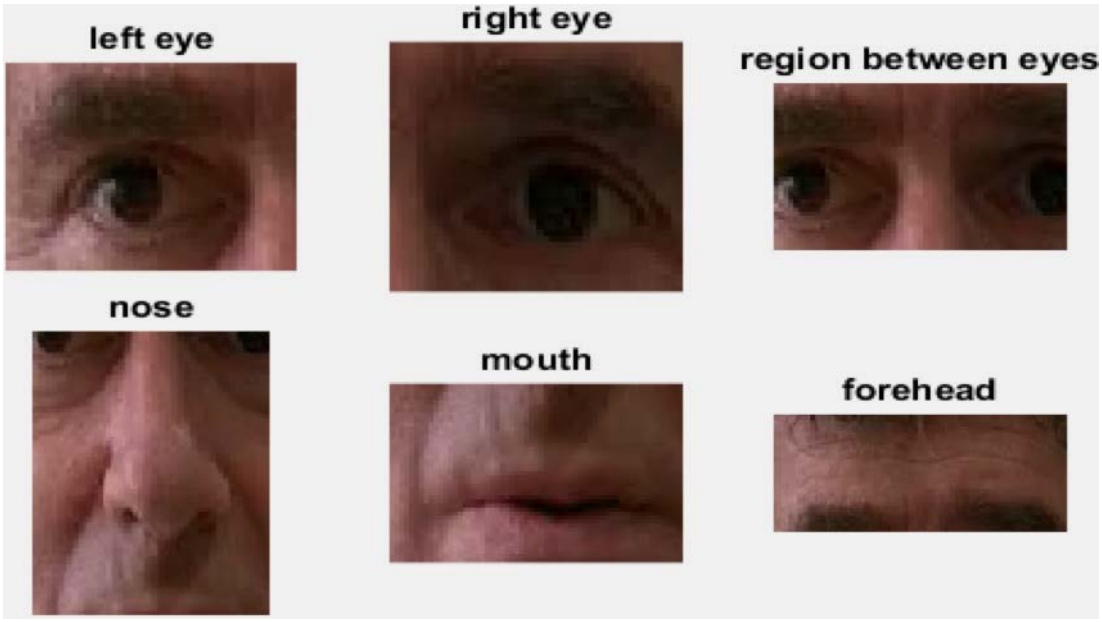

Figure 6 Illustration of the six salient facial regions of interest (ROI): left eye, right eye, nose, region between eyes, mouth and forehead

2) **CNN Features:** Our CNN face representation is based on the VGG-face model [16], which is a 16-layer convolutional neural network (CNN) model trained with 2.6M facial images of 2.6K people for face recognition in the wild. We use this model for feature extraction by employing the output of the sixth Fully Connected layer (FC6) as the facial signature. This layer outputs a 4096-dimensional feature vector.

3) **Geometric Features:** Another feature representation deals with the shape and location of the facial landmarks (e.g. mouth, eyes, eyebrows, nose, and chin). Different facial expressions correspond to different shape deformations of the facial landmarks. The location of the fiducial points was chosen according to the facial model proposed by [26] and is shown in Figure 7. These landmarks are transformed and fitted with the same alignment used for face registration. An alternative is to use the fiducial points coordinates as features in the classification process, but this representation achieves deficient performance, as it is not able to capture the variations between different individuals. For increasing the discriminative power of the feature set, we compute geometric features, which may be represented by segments, perimeters, or areas of the figures formed by the fiducial points. Following the works in [29] and [30] we obtain a set of features including: Euclidean distances, angles and curvatures between fitted facial landmarks, followed by applying a normalization step. For example, the set of extracted features include but are not limited to: mouth and eyes aspect ratios, lower and upper lips and mouth corners' angles, nose tip- mouth corner angles, eyebrow slope, mouth corner and mouth bottom angles, and the curvature of lower-outer and lower-inner lips.

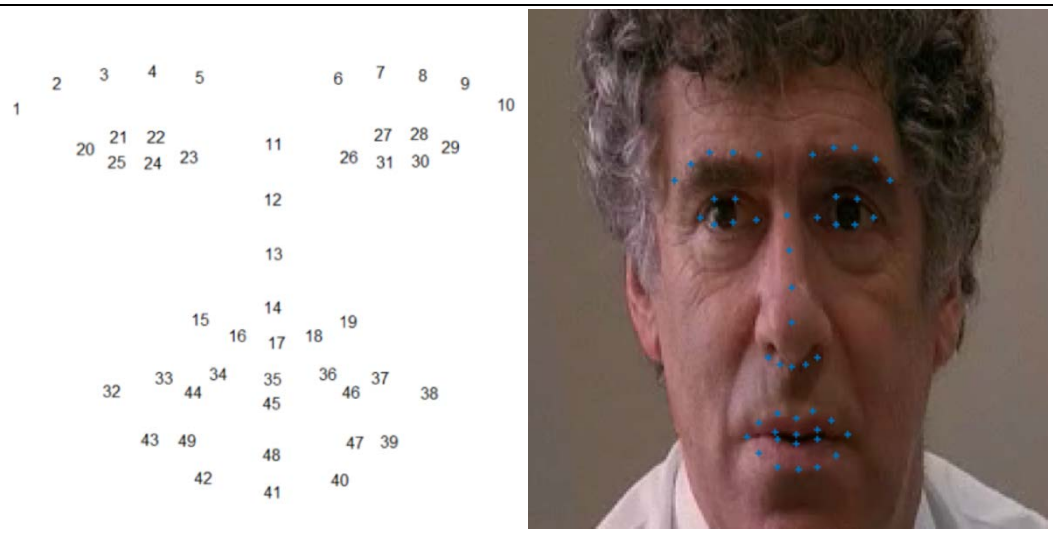

Figure 7 Facial landmarks provided by [26].

4) **Audio Features:** We utilize the speech analysis openSMILE toolkit [31] for audio features extraction. This popular and widely used library extracts features that capture both voice quality and prosodic characteristics of a speaker. We follow the audio feature extraction as explained in [32]. The set of audio features used in this study consists of: 34 energy & spectral related low-level descriptors (LLD) × 21 functionals, 4 voicing related LLD × 19 functionals, 34 delta coefficients of energy & spectral LLD × 21 functionals, 4 delta coefficients of the voicing related LLD × 19 functionals and 2 voiced/unvoiced durational features. The details for the LLD are included in Table 10. The functionals computed on the LLD include: arithmetic mean, standard deviation, skewness, kurtosis, quartiles, quartile ranges, percentile 1%, 99%, percentile range, position max./min, up-level time 75/90, linear regression coefficient, and linear regression error (quadratic/absolute).

Low Level Descriptors (LLD)	Audio Features
Energy/Spectral LLD	PCM Loudness MFCC [0-14] log Mel Frequency Band [0-7] Line Spectral Pairs (LSP) frequency [0-7] F0 F0 Envelope
Voicing related LLD	Voicing Prob. Jitter Local Jitter consecutive frame pairs Shimmer Local

Table 4 Audio features: low level descriptors

3.2.3 Feature Encoding and Video Modelling

Video Modelling: In this work, we adopt the usage of Fisher vectors for encoding and clustering different low-level features for each modality. The features are not only pooled from one still image,

instead they are pooled from all the frames across a face track. As suggested in [28], we use video-pooling, where we compute a single fisher vector over the whole face track by pooling together low-level features (e.g. DSIFT, or CNN features) from all facial images in a track. This kind of representation has many advantages comparing to still image based representation for several reasons: (i) it encodes the spatio-temporal information in a face track, (ii) it captures the motion of the face over time which leads to a better description of the different low-level features; and (iii) it dramatically reduces the dimensionality of data by producing a single discriminative descriptor for a video.

Fisher Vector Representation:

The pipeline for Fisher vector encoding typically starts with extracting a set of features (e.g. DSIFT, geometric features etc.), and then aggregates the large set of feature vectors across all frames in a track into a high dimensional Fisher vector which is better suited for linear classification. This is achieved by fitting a parametric generative model such as Gaussian Mixture Models (GMM) to the features. GMM can be referred to as a probabilistic visual vocabulary. The next step consists of encoding the gradient of the local descriptors log-likelihood with respect to the GMM parameters. The GMM parameters are estimated on a large set of local descriptors using the Expectation Maximization (EM) algorithm to optimize the log-likelihood. In Fisher vector computation, the covariance of the GMM is assumed to be diagonal and only the derivatives with respect to Gaussian mean and covariance are considered. This leads to a vector representation that obtains the average first and second order difference between the features and each of the GMM centres:

$$\Phi(k)^{(1)} = \frac{1}{N\sqrt{w_k}} \sum_{p=1}^{N} \alpha_p(k)\left(\frac{X_p - \mu_k}{\sigma_k}\right)$$

$$\Phi(k)^{(2)} = \frac{1}{N\sqrt{2w_k}} \sum_{p=1}^{N} \alpha_p(k)\left(\frac{(X_p - \mu_k)^2}{\sigma_k} - 1\right)$$

Where w_k ,μ_k ,σ_k are the GMMs weights, means and diagonal covariance. $\alpha_p(k)$ is the soft assignment of the p-th feature x_p to the k-th Gaussian component. Fisher vectors dimensionality is 2Kd which depends on the number of the GMM components (K), and the dimensionality of the used set of features. Then, the Fisher vector ϕ is computed by stacking the differences (the assignment of the local features to the first and second differences of GMM centres): ϕ = [Φ (1) $^{(1)}$, Φ (1) $^{(2)}$ ...., Φ(K) $^{(1)}$, Φ(K) $^{(2)}$].

A Fisher Vector representation has many advantages: (I) it is a generic representation which combines the benefits of generative and discriminative approaches, (ii) it can be computed using a small number of parameters (GMM parameters), (iii) more importantly, it is efficient, and it shows a significant benefit when used in combination with linear classifiers such as linear-SVM [33].

3.3 Fusion approaches

In this section, we present the **two fusion approaches** employed in our study of multimodal fusion framework: feature level fusion based on information gain; and score level fusion, improved by means of a genetic algorithm. We propose a framework for multimodal emotion recognition, which **combines different modalities in a hierarchical and collaborative fashion**, using both early and late fusion schemes. These two techniques aim to maximize the benefit of different modalities in emotion recognition. In the rest of this section, first, we introduce our approach by explaining how information gain and Fisher vector representation are involved in early level fusion. Then, we

describe our method of collaborative late level fusion that captures the performance of each modality per emotion to enhance the final decision-making.

3.3.1 Feature Level Fusion

In our study, we apply various feature extraction and representation techniques for different modalities. Accordingly, their data comes from diverse input channels. Therefore, each modality has its own distinct feature distribution properties. However, a multimodal fusion and feature learning method can be used to capture the correlations between these modalities in real word data, by employing a feature level representation. As a result, similarity in the representation space, must reflect the similarity in corresponding concepts. For example, speech and facial images are correlated in the real world when people express their emotions. People often tend to speak loudly when they are angry, or they use a certain tone of voice accompanied by a facial expression to indicate their affective states. We use Fisher vector encoding to map the extracted features into a common space, for achieving a higher layer feature description, which shares similar statistical and discriminative properties. Thus, different modalities are projected into one domain by means of fisher vectors, enabling and supporting feature concatenation. The newly obtained fisher vector based representation is independent of the input modality, opposite to the low-level features which are modality-dependent. Furthermore, the proposed data representation is useful at capturing the non-linear relationships between the different modalities employed in our work.

3.3.2 Feature Level Fusion Based on Information Gain Principles

For optimizing the feature level fusion of different modalities and selecting the best combination among the possible ones, we used measures from information theory, as the Kullback-Leibler (KL) divergence [20], which is useful at measuring the distance between two probability distributions (PDF). In our framework we aim to minimize the distance between the PDF of the true labels, denoted with Y and the PDF of the predicted labels for each modality (X_k , k ∈{1 ..., $\mathbf{n}_{mod}$ }), obtained using a classification algorithm on top of the modality k features.

$$KL(X_k||Y) = \sum_{i=1}^{N_{lab}} X_k(i) log \frac{X_k(i)}{Y(i)}$$

Where N lab is the number of labels, X k is the PDF of predicted labels for the k modality, and n mod is the number of input modalities denoted by several types of features, both visual and audio. By minimizing the KL divergence, we aim to obtain a PDF as close as possible to the ground truth PDF, increasing in this way the performance accuracy of our emotion recognition framework. As the KL divergence is not symmetric, we employ in our work the symmetric version [34], for obtaining a general framework, which is not affected by the order of the modalities in the fusion process:

$$I(X_k, Y) = \frac{KL(X_k||Y) + KL(Y||X_k)}{2}$$

Furthermore, the set of modalities which are fused at the feature-level are selected by minimizing the KL divergence between the PDF of the true labels and the PDF of the predicted labels using a set of fused modalities, achieving in this way a result as close as possible to the expected one:

$$\underset{k,j}{\operatorname{argmin}} I(\{X_k, X_j\}, Y), k, j \in \{1, \ldots, n_{mod}\}, k < j$$

3.3.3 Score Level Fusion

In our work, we observed that emotional states are more dominant depending on the existing modalities, e.g. some of them are visual prevailing, while others are stronger displayed through the audio modality. As modalities can be complementary to each other and display varying performance characteristics across emotions, we take advantage of this aspect for **predicting emotional states in a collaborative manner at the decision level**. We apply this scheme in two stages, first we learn classifiers for each single modality separately, and then we combine the scores of specific modalities at the decision level.

In the first stage, each modality classifier is regarded as an expert model due to its distinctive performance in emotion prediction. In this phase, we take advantage of the best fused modalities obtained using the information gain principles presented in the section of multimodal learning. In addition, we also use classification techniques for each modality or fused feature vector before feeding it into the decision level algorithm, as different classifiers are better fitted for specific modalities [35]. Then, we apply a weighting scheme that takes into consideration the performance of each modality with respect to each affective state. The final decision is obtained using a weighted-sum of the prediction given by each modality. For optimizing our results, we employ a genetic algorithm (GA) for assigning weights to each modality score for each affective state.

For the aforementioned reasons, we applied a re-weighting per modality and per emotion as a hyper-parameter search over the model prediction scores for each emotion. This optimized search algorithm adjusted the parameters to produce a collaborative and complementary scheme. Accordingly, GA learns the weights of the final decision for the modalities combination and their predictions. The search space S of GA depends on the number of modalities fed into it: $\mathbf{n}_{lab}$, and the number of predictions $\mathbf{n}_{mod}$ for each modality which is fixed to 7 in our case (the number of basic emotions and the neutral state). Therefore, the search space matrix *S* has [$\mathbf{n}_{mod} \times \mathbf{n}_{lab}$] dimensions. Prior to learning the weighting scheme of the selected modalities, we considered lower and upper bounds constraints to avoid over-fitting the given modalities by GA. We use the following constraints to regularize the learning during the weight parameters search:

$$0 \leqslant S(k,i) \leqslant 1, \&k \in \{1,\ldots,n_{mod}\}, i \in \{1,\ldots,n_{emo}\}$$

3.4 Evaluation

3.4.1 Dataset

The dataset chosen for the training and evaluation of our framework is the Acted Faces Emotion in the Wild (AFEW). This dataset is a dynamic temporal facial expressions data corpus consisting of close to real world environment extracted from movies. A complete description of this database was included in [36]. As previously mentioned, there was a lack in database within MaTHiSiS, and additionally, the collected individual sensorial input data from some use cases did not allow us to study or train the aforementioned models due to the resulted dominance of one specific label which is Engagement. However, recently AFEW dataset emerged as a benchmark dataset to study multimodal emotion recognition in scientific community. Extensive details and statistics about the collected dataset within MaTHiSiS is described by deliverable D4.3 in [25].

Figure 8 Example of still images of affective states and a face track from the AFEW dataset

3.4.2 Evaluations Metrics

In our experiments, we take into consideration several evaluation criteria: (i) Accuracy, which is the number of correctly classified video samples; (ii) Confusion Matrix between the ground truth and the predicted emotion labels and (iii) Symmetric KL-Divergence, where we aim to minimize the symmetric KL-divergence between the predicted labels and the true labels. We train our proposed approach on the train set and test it on the validation set.

3.4.3 Unimodal Experiments

Firstly, we apply the evaluation metrics for each representation separately on the AFEW validation set. These experiments aim to show the baseline performance of distinctive features for both visual and audio modalities, which are presented in Table 5. The best results are obtained for the visual modality, where CNN appearance based features are slightly better than the baseline results. Another interesting finding is represented by obtaining an improved accuracy of audio features when encoded with Fisher vectors in comparison to the raw audio features. As shown in last two rows of Table 5, this gain in the performance by almost 6% is significant.

Modalities	Features	Accuracy
Visual	FV on DSIFT	39.4%
Visual	FV on ROI DSIFT	39.2%
Visual	FV on CNN features	40.0%
Geometric	FV on hand crafted geometric features	32.8 %
Audio	FV on audio features	36.4 %
Audio	Raw audio features without FV	30.8 %

Table 5 Performance of individual modalities on AFEW validation set using linear SVM classifier

3.4.4 Multimodal Emotion Prediction Feature Level Fusion

We first encode the low-level features of audio and visual modalities using a Fisher vector representation. To such an extent, we obtain a general representation of each modality that shares similar distribution properties. Next, we concatenate the Fisher vectors of pair modalities and then perform the classification task using linear Support Vector Machines (linear-SVM). In case of concatenating the Fisher vectors of all modalities, the accuracy on the AFEW validation set is 45.6%. In addition, using IG principles, based on minimizing the symmetric KL-divergence between the

predicted labels of concatenated modalities and the ground truth labels of the test samples, we selected the best combination of features to concatenate, followed by the emotion prediction task. This leads to an overall accuracy of 47.5%, for a reduced set of modalities composed of (CNN, geometric, DSIFT and audio). Figure 9 shows the confusion matrix of affective states corresponding to this approach. However, fusing all the modalities into one feature vector is less efficient for the classification task and also slower in comparison with to the following scheme of score level fusion, which is based on the fusion of the best pair modalities.

Figure 9 Confusion matrix of the AFEW validation set for the IG based feature level fusion.

Dataset Influence: in the AFEW dataset, there are a number of videos for which it is very hard to decide their emotion label only from the visual information. For example, we noticed that, facial expressions, in many videos, labelled as surprise have been classified as an angry emotion by several human annotators, observation also supported by [37]. There- fore, we need more contextual and complementary information to enhance the accuracy and to correctly classify these ambiguous videos. Thus, the audio modality represents one way to boost up the performance of the classification task by adding contextual information. In addition, we observed that in both fusion schemes, employing distinctive features and modalities led to a better accuracy. In Table 5, the performance of separate modalities is reported, (e.g. face and audio accuracies are 39.4% and 36.4% respectively). Accordingly, Table 6 illustrates the performance for feature level fusion, where we notice that the fusion of visual and audio modalities increases the performance to 43.3%, mainly due to the complementarity of the two channels.

Fusion	Modalities	Sym-KLDV	Accuracy
FLF	ROI DSIFT and Geometric	0.2622	43.6%
FLF	Audio and DSIFT	0.2626	43.33%
FLF	Geometric and DSIFT	FLF 0.3244	40.6%

Table 6 Performance of feature level fusion on concatenated pair modalities of AFEW validation set.

As we aim to investigate the advantages of a hierarchical fusion scheme, we apply the information gain theory based on minimizing the KL-divergence for deciding the best pair of modalities to be combined in feature level fusion. In Table 6, the three best pairs of modalities are included, obtained by concatenating the FV of the following features: (i) ROI DSIFT and geometric features, (ii) audio and DSIFT features, and (iii) geometric and DSIFT features. The KL- divergence and the obtained accuracy on the AFEW validation set, are shown in Table 6. We notice the increase in the performance over the unimodal results in Table 5in all cases, which proves the benefits of both feature level fusion and of the Fisher vector representation.

Score Level Fusion: Following the feature-level fusion step, we feed the obtained pair modalities predicted scores into late level fusion, and searched for the best weights to fuse them. As emotions are more dominant depending on the audio or visual modalities, score level fusion aims to breakdown the fusion into this level, where we assign weights per-modalities and per-emotion. For achieving this purpose, we employ two approaches: (i) firstly we use as weights of each modality the diagonal elements of the confusion matrix; (ii) the second technique uses GA for searching the best weights to fuse the given modalities. In the first case of using the performance based weights, the overall accuracy is 44.4%. However, in the second case, we apply a genetic algorithm as an optimization search algorithm, using 5-fold cross validation. Figure 10 depicts the score level fusion approach together with the weights per-modality and per-emotion in the best performing case. The GA-optimized search resulted in an enhanced performance with an average accuracy of 48.9%. The results obtained in both cases are shown inTable 7. In comparison to the feature level fusion and the performance based weights late fusion, the genetic algorithm outperformed both approaches, leading to a better fusion model.

Score Level Fusion	Accuracy
Genetic Algorithm Based Fusion	48.9%
Performance Based Weights Fusion	44.4%

Table 7 Score level fusion (SLF) of pair modalities in Table 3 on AFEW validation set.

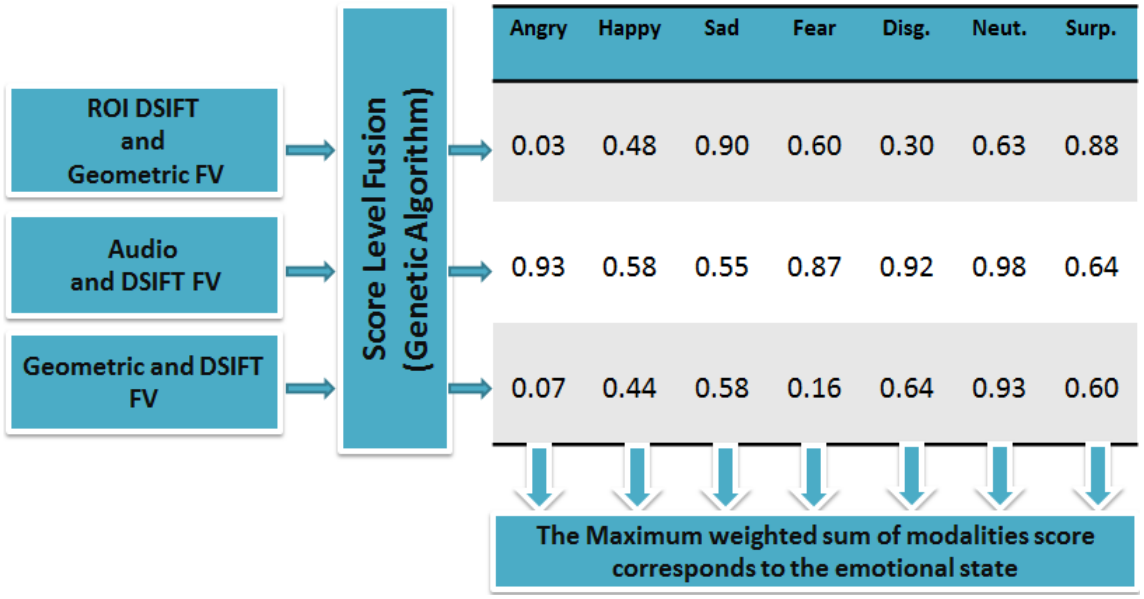

Figure 10 The resulted modalities and features from feature level fusion by FV and IG, and the weights per-modality and per emotion obtained by score level fusion using GA.

In addition, the best weights of among the 5 folds gave an even better performance, obtaining an accuracy of 53.06%. Figure 11 contains the normalized confusion matrix for the validation set obtained using the best weights of the score level fusion. When compared with the confusion matrix for the best feature level fusion, we notice a substantial improved performance for several emotions (angry, sad and neutral).

Therefore, we can notice the advantages of our multimodal learning scheme for combining the feature level and the score level fusion in a hierarchical manner based on IG principles and GA optimization. Additionally, score level fusion has the advantage of re-weighting existing modalities to benefit from their individual expertise and performance on specific emotions.

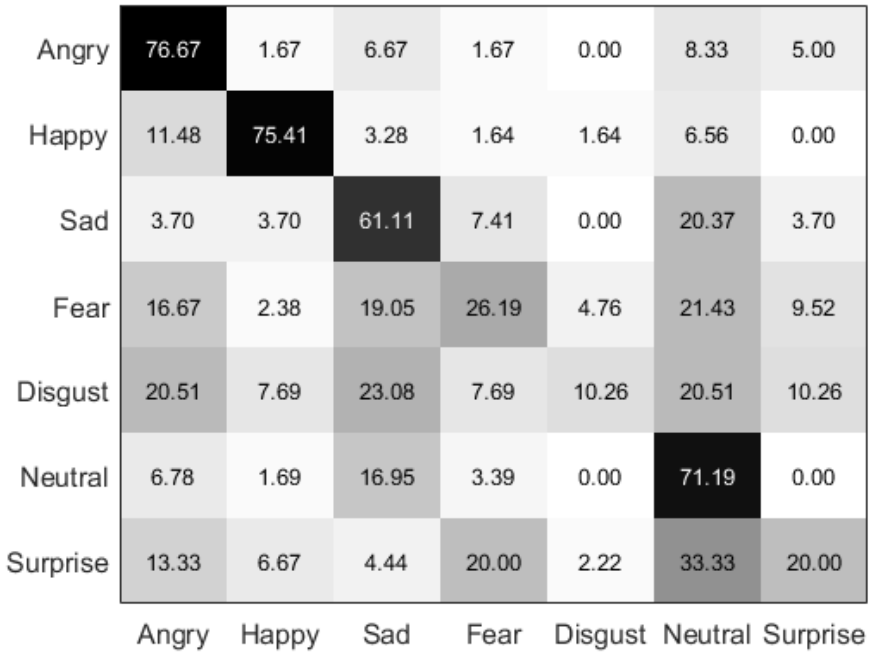

Figure 11 Confusion matrix of the AFEW validation set for the best score level fusion

3.5 Multimodal fusion in MaTHiSiS platform

The fusion scheme in MaTHiSiS platform is shown in Figure 12. As shown in the figure, there are different sensorial components in MaTHiSiS such as face, audio, mobile and eye gaze. In addition, there is a component based on the interaction of learners with the learning materials and platform agents called (IPA). Subsequently, these different input channels will be fused according to their availability and relevance in the learning environments and the use cases. The currently functional and running scheme is based on equal weighted score level fusion. This scheme was proven advantageous and better than feature level fusion using public datasets. Further plan is to build score level fusion, which is based on weight per modality and per use case, as the contribution of each modality varies according to the use cases. Weights can be obtained by training an algorithm such as Genetic Algorithms or using experts' recommendation per use cases.

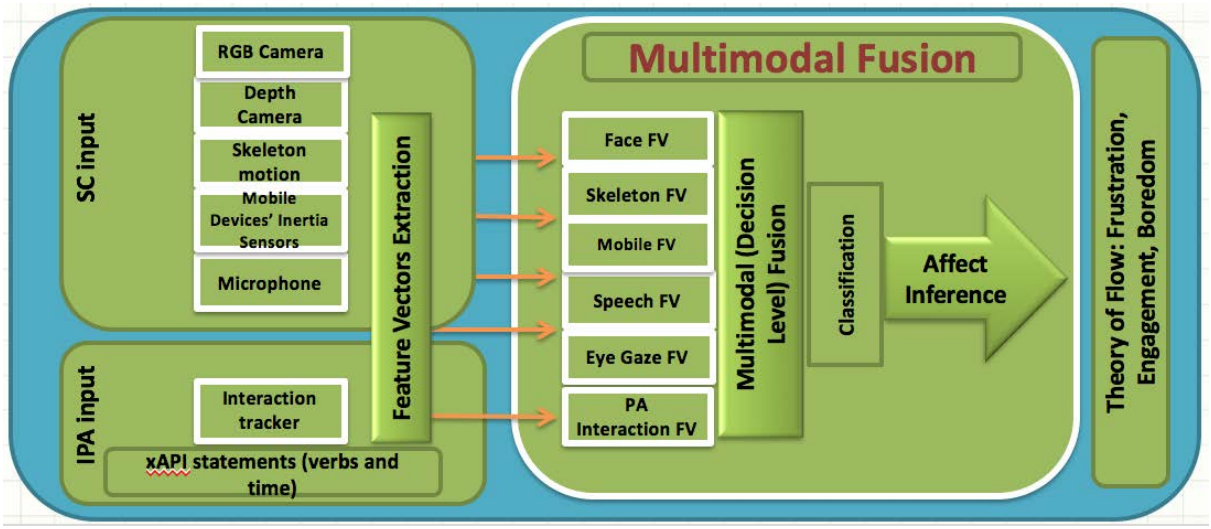

Figure 12 Multimodal Fusion and AIRlib in MaTHiSiS

Currently, this method has not been implemented in MaTHiSiS platform completely. However, the multimodal fusion performed follows a late fusion approach, since this method reflected results that are more promising. In the version integrated in the release to be used during the assisted pilots, a simple fusion of labels is performed. Moreover, this component has been implemented following a flexible approach that allows the calculation of the current affective state of the learners using a non-fixed number of modalities (including SC modalities and IPA information).

This component belongs to the Decision Support System or DSS (for further information, [24] may be consulted). In order to access this functionality, two POST methods have been implemented as part of the corresponding RESTful API. These methods take as input a JSON object. A whole description of these methods is included in https://gitlab.atosresearch.eu/ari/mathisis/wikis/air-lib-api. The information provided by the IPA component is shown in [25] (Section 2.2.1) whereas in the case of the SC, data is provided using a similar format. The SC communication JSON object is described as follows:

```
{
"affect_label_probs": [
[
{Boredom probability},
{Engagement probability},
{Frustration probability}
```

```
],
…
],
"affect_labels":[
{Affect label with highest probability per SC modality},
],
"features":[
[
],
],
"learner_id":{Learner identifier},
"sensor_num":{number of modalities},
"sensors_type":[
{name of the modalities}
],
"session_id":{Session identifier},
"timestamp":{Timestamp of the interaction}
}
```


Furthermore, additional tests were being conducted at time of the submission to integrate the weighted multimodal fusion approach (as described in Sections 3.2.3 and 3.4.4 and Figure 10). In the context of MaTHiSiS platform, the weights will be established per modality and per use case. This weighted method will allow a personalised fusion of modalities considering patterns of behaviours of each use case, analogous to the approach validated previous, which takes into consideration the performance of each modality with respect to each affective state. In addition, MaTHiSiS approach is based on the Theory of Flow representation, as described in [37] (engagement, boredom, frustration).

4. Performance calculation

User Performance is a local assessment of the speed of successful task completion. This is a descriptive attribute of user Skill and is dependent on the LA difficulty involved. In MaTHiSiS at each moment there can be numerous LA's involved in a learning activity. Thus, to simplify calculations still presenting comprehensive metrics, an average of the difficulty levels is taken. For the rest of this document, this is represented by ASLAW (Average Smart Learning Atom Weight). Performance gives a quantifiable rate for those learner achievements. Performance can be used to compare user achievement between activities, even in different scenarios. Due to its dependency on LAs, it cannot be used to compare two users in two different scenarios - but it can be used to historically compare the ability of each user in their own scenario. (It is section 5.2 that focuses on comparisons among learners.)

For example:

User X has a Performance of 70 in activity A and user Y has a Performance of 50 in Activity B. This does not mean User X is more intelligent or capable than User Y. However, it does mean that User X is performing better and achieving more of A than User Y is achieving in B. This comparison reflects on the qualitative experience of User X, where User X is more confident and capable in A than User Y is in B.

$$Performance = \frac{SLAW}{Response\ Time} * User\ Response\ Value$$ **Equation 1**

Response Time is the time the user takes to respond to a challenge. Specifically, the total time the user spends responding to answers, inclusive of correct, missed and incorrect responses.

4.1 Temporal performance

Local comparisons are used to visualize the current user performance to previous steps in order to determine if the user is performing better than moments ago, or if their performance degrading. This type of comparison is useful in making adaptive adjustment while the user is still interacting with the learning experience. Temporal user Performance can be evaluated in real-time for the current interaction and then a near point comparison could establish if the user's performance is degrading or improving. This comparison is restricted to interactions with the same average LA difficulty from the same LG, making it less useful to compare performance differences between activities from one LG's but with two various levels of average LA Difficulty. For cases where a mix average LA difficulty comparison is required, the Global comparison method described afterward is used.

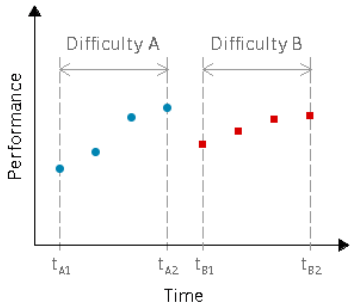

Figure 13 Local comparisons can be done within similar Difficulty levels

4.2 Global performance

A global comparison is used to compare user performance between sessions. It is different from a local comparison because it facilitates the comparison between user sessions of mixed difficulty and SLAs. Global comparisons could be used to determine two evaluation types.

1. Performance comparisons
2. Response Accuracy comparison

Performance comparison

Global performance comparison can demonstrate an improvement or a decline in the user performance from one ASLAW to the next. An appropriate index for this comparison is Overall Performance. Overall Performance has restrictions on the SLA sample, if a comparison is being made both sides of the comparison are required to use the same SLAs. Between two sessions we could have.

$$Overall\ Performance_{ALA_1 ALA_2} > Overall\ Performance_{ALA_1 ALA_2} \quad \textbf{Equation 1}$$

Response Accuracy comparison

Response Accuracy can be useful feature for detecting user affect state and determining when a user has reached a threshold accuracy. This threshold can be used to assess when a user has shown significant correct responses to challenges. When a user reaches the required accuracy threshold, they have acquired that Difficulty level in Skill. This threshold can be set differently for LAs in varying Learning Goals.

A comparison can also be used to evaluate the different levels of ASLAW with each other or between users to determine which user has progressed further in a collaboration scenario.

4.2.1 Reporting Values

In any comparison, either local or global there may be a difference in sample quantity. To achieve a robust evaluation, sample quantity must be taken into account for any reporting and comparison. For example, for a similar confidence level of 95% a sample quantity of only 50 values with has a confidence interval of ± 3.24 while a sample size of 5 would nearly triple that interval to ± 10.24.

Range for the true population mean for 5 samples: 90 ± 10.24
and for 50 samples only 90 ± 3.24.

Therefore, with more samples, the precision of the assessment is much higher and evaluation in turn becomes more representative. This demonstrates the importance of always reporting any evaluation or comparison with confidence intervals.

The formula for confidence interval is:

$$Confidence\ Interval = z^* \frac{\sigma}{\sqrt{n}}$$ **Equation 2**

Where σ the standard deviation, and n is the sample size.$z^*$is the confidence coefficient and can be looked up from Table 8 for the desired level of confidence.

Confidence Level	z^*
0.7	1.04
0.75	1.15
0.8	1.28
0.85	1.44
0.9	1.645
0.92	1.75
0.95	1.96
0.96	2.05
0.98	2.33
0.99	2.58

Table 8 Confidence level to Confidence Coefficient lookup table

4.3 Storing performance values in database

A function has been developed that calculates the temporal performance of the learner in a set time interval of 5 seconds. This function can also be executed on demand when needed (for example when new interaction is received by the system from the learner). This interval and on-demand triggering of the function insured that there is a steady and up-to-date level of performance available in the system at all time. Having the system produce performance values on interval insures there is sensitivity in the system for periods of learner absences. The performance calculation for the learner is stored in the central MaTHiSiS Mongo database.

Performance	Description
_id	Internal MaTHiSiS ID used for learner
Session ID	Internal MaTHiSiS ID used for the session
Learner ID	Internal MaTHiSiS ID used for the learner ID
Current LA ID	All active LA Id's as an array
Time	Time that event happened (Unix Timestamp)
Response time	Time between interactions that indicate start of an activity that asks the learners to respond to something and the actual learner response (in ms)

slai_weight	An average of multiple SLAs (per the LA executed)
Score	Score value calculated
Interaction Type	Combination of Interaction type if PA asked a question and also learner reaction. [PA asked question], [passed, fail]

Table 9 Required data to calculate performance and update performance table

Performance	Description
_id	Internal MaTHiSiSID used for learner
Session ID	Internal MaTHiSiS ID used for the session
Learner ID	Internal MaTHiSiS ID used for the learner ID
Current LA ID	All active LA Id's as an array
Time	Time that event happened (Unix Timestamp)
Response time	Time between particular interactions that indicate start of an activity that asks the learners to respond to something and the actual learner response (in ms)
slai_weight	An average of multiple SLAs (per the LA executed)
Performance	Temporal performance, a value between 0-1
Confidence	Performance value confidence rating as a numeric percentage
Score	Legacy score value calculated
Interaction Type	Combination of Interaction type if PA asked a question and also learner reaction. [PA asked question], [passed, fail]

Table 10 Performance database structure

5. Learning analytics

This section describes how the information generated during the execution of the learning experiences of the learners is used to produce insights and progress reports for the tutors and learners or their parents/caregivers.

In principle, when building learning analytics there are different dimensions that need to be considered[41]:

- **Stakeholder side**: what the users need, what information we have. Special attention should be paid to the fact that Learning Analytics is about learning and the learning perspective should be always in the centre along with the tutors and learners.
- **Technological side**: the MaTHiSiS cloud; including the data residing in the MongoDB, the backend APIs and frontend interface of the MaTHiSiS platform.
- **Purpose of the Learning Analytics**: it is about reflexion and prediction: reflect about the learner or class progress and predict through the dashboards. In MaTHiSiS we could try to respond through the dashboard to a question such as "*Do learners actually learn using MaTHiSiS and are they doing it in an efficient way?*". Insights including the time to learn, their affective states in combination with their performance information and meta cognitive skills such as collaboration style could help into trying to respond to such a question.
- **Different type of data inputs**: multimodal data, xAPI data with the learners' interactions with the different devices, effective states, learner performance, personalised learning graphs.
- **Privacy and ethics issues**, which are managed through the ethical task in MaTHiSiS and is outside the scope of this deliverable.
- **Competence side:** how to well-present the data. It is very important how data can be mashed up and visualised; a single dataset may be visualised in many different ways and have totally different meanings.

5.1 Learning analytics component

The Learning Analytics component is responsible for performing analytics queries on the MongoDB repositories that reside in the MaTHiSiS cloud and display the information in the form of an intuitive dashboard for the tutors and learners.

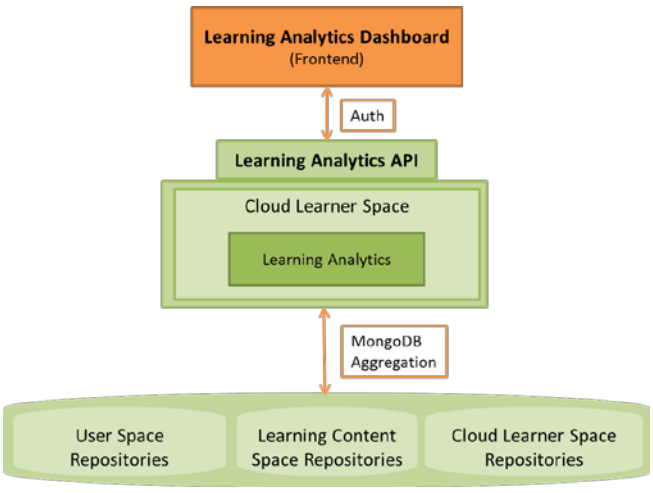

Figure 14 Learning Analytics component architecture

Learning Analytics is composed by a backend component that exposes an API to the frontend (Figure 14). The main functionalities include the execution of analytics queries on the historical data residing on the MaTHiSiS repositories utilising the power of MongoDB's Aggregation operations [43] that allows performing a variety of operations of grouped data from multiple documents in order to return a single result.

Currently, the following documents are accessed from the MongoDB repositories:

- **Venues**: contains the information of the learning environments
- **Classrooms**: contains the information regarding the classrooms that belong to the learning environments.
- **Users**: contains the information of all types of users (tutors, caregivers, learners) as well as learner profiles in the case of learners.
- **LearningSessions**: contains the information regarding the learning sessions of each learner and learning experience
- **lcsLearningGraph:** contains the information of a leaning graph
- **usLearningGraphInstance_rtm**: contains the runtime historical information of the learners personalised learning experiences together with the weights.
- **dss_affective_states:** contains the historical and runtime affective state data of the learner during a learning experience. This information is recorded at the end of the multimodal fusion that may happen many times at the course of a learning session.

Learning Analytics also includes a frontend component, which resides within the MaTHiSiS common web frontend. This is where the MaTHiSiS Learning Analytics dashboard is configured which consumes the data through the Learning Analytics API.

5.2 MaTHiSiS Learning Analytics dashboard

The Learning Analytics dashboard resides on the web frontend of the MaTHiSiS platform. An initial version was available during the Driver pilots with the pre-alpha version of the platform. At the moment of writing this document, a more elaborated version is available with the Alpha version of the platform ready to be used during the Assisted pilots. This version, which was created based on the data currently generated by the MaTHiSiS platform, will be further enriched with the Release Candidate planned before the Real Life pilots taking into account the feedback from the pilot partners and the new information generated by the platform, such as the learner´s performance calculation.

The users who are supposed to be using the Learning Analytics dashboard are in principle the tutors and leaners. For the case of supervised learners, the caregiver may access the dashboard for their assigned learners. Therefore, the dashboard has been designed with these types of users in mind.

The visualisations included in the dashboard are built using the Highcharts [42]. All visualisations are interactive, allowing removing or adding lines from the chart by clicking on the corresponding legend. Another useful feature is the possibility to zoom in parts of the visualisation, which is very useful in case we need to get for example detailed hourly insights for a few days.

In order to access the dashboard, the user needs to enter the login information and then navigate to the "Learning Experience Supervisor" and then to the "Analytics Dashboard". In the following sections the different versions of the dashboard according to the role of the logged in user is detailed.

5.2.1 Tutor dashboard

The tutor´s dashboard is the most elaborated, since there is a lot of information that might be of interest to the tutor. The dashboard is divided into two core sections; the Classroom overview and Learner overview. As a first step, the tutor needs to select the Learning Environment that wants to consult and also the specific classroom for which to see the overview. The tutor can only see and access the Learning Environments and Classrooms that he/she is assigned to. After the selection of the classroom, the dashboard is generated.

5.2.1.1 Classroom overview

On a classroom level, it is possible to view the information based on all learning experiences used in the classroom or by selecting only learning experience- specific analytics. The following visualisations are available to the tutor:

Learning progress overview

The Learning Progress overview visualisation shows to the tutor at a glance the average progress of the classroom. It displays a line chart showing the average uptake on the presented learning contents of all the learners of the classroom per learning experiences.

A learning experience is represented by a personalised graph, made by vertices and their relations. Vertices represent the learning content components, and their relations are represented by directed edges. In the personalization process of the graph, these vertices are assigned with a scalar weight that represents the uptake of the learner (whom the graph is personalised for) for that vertices. In the general case of learning progress overview, where a classroom is selected but no specific learning experience is selected, the graph shows the average weight of all the vertices in a learning experience, along the time, for all learning experiences where all learners in the specific classroom take part. This way, it shows how knowledge of the group of learners in one classroom is progressing in time, for all learning experiences. The time granularity can be changed in order to view information per hour, day or month.

Graph in Figure 15 provides an average for the classroom.

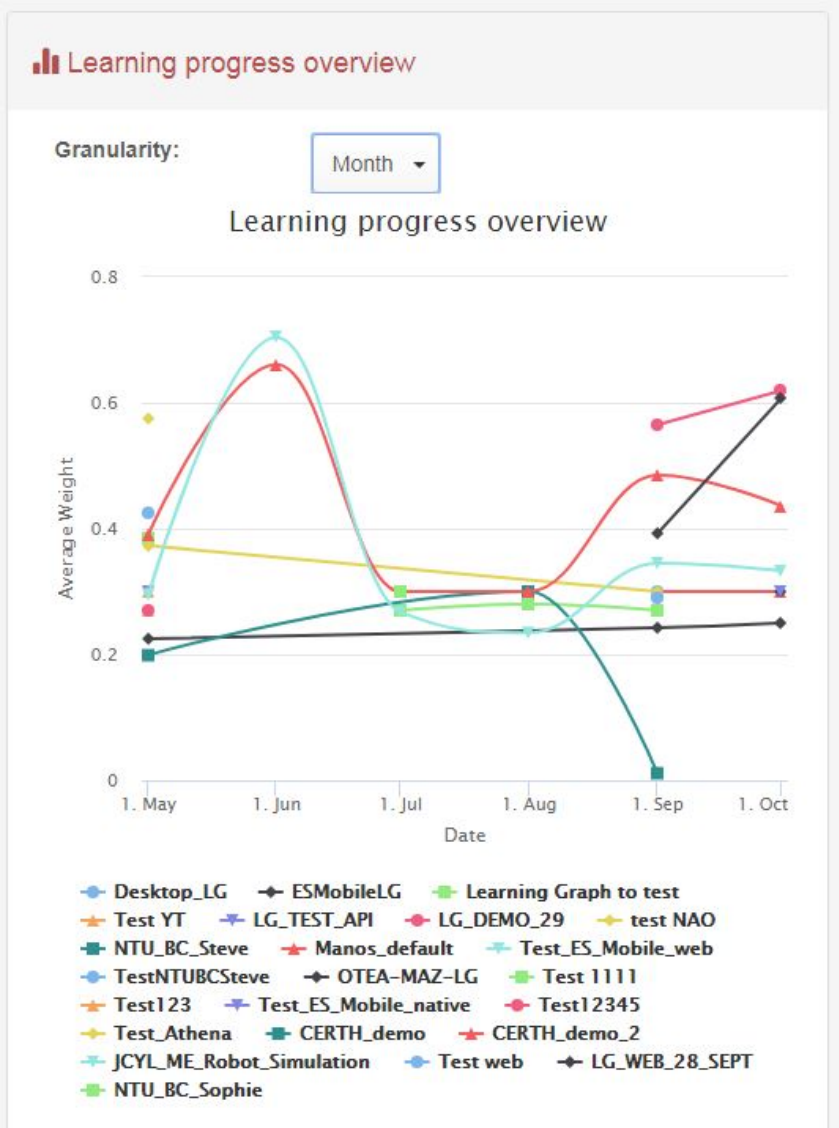

Figure 15 Learning progress overview for a classroom

There is also the option to select a specific learning experience, and check only the progress of the classroom on a concrete learning experience as displayed in Figure 16.

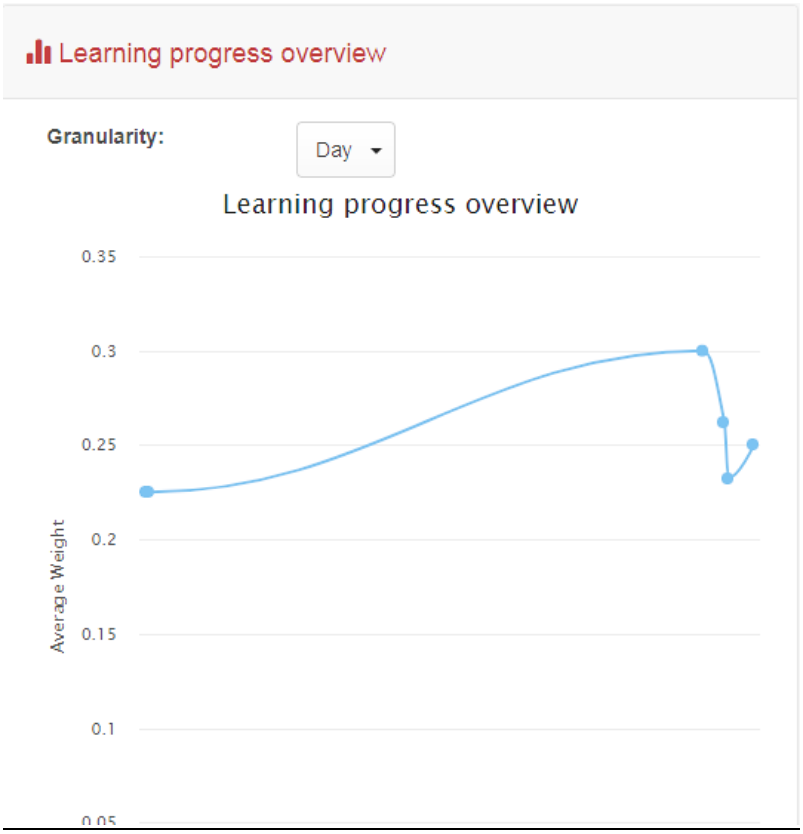

Figure 16 Learning progress overview for a specific learning experience (ESMobileLG).

Effort overview

The effort overview graph (Figure 17) shows the overall time spent on each learning experience at a classroom level to evaluate the effort employed to learn specific subjects. The time is calculated considering when the first learning session for each learning experience is launched and by accumulating the duration of the learning sessions until the least learning session is ended. The time granularity can be changed to view information per hour, day or month.

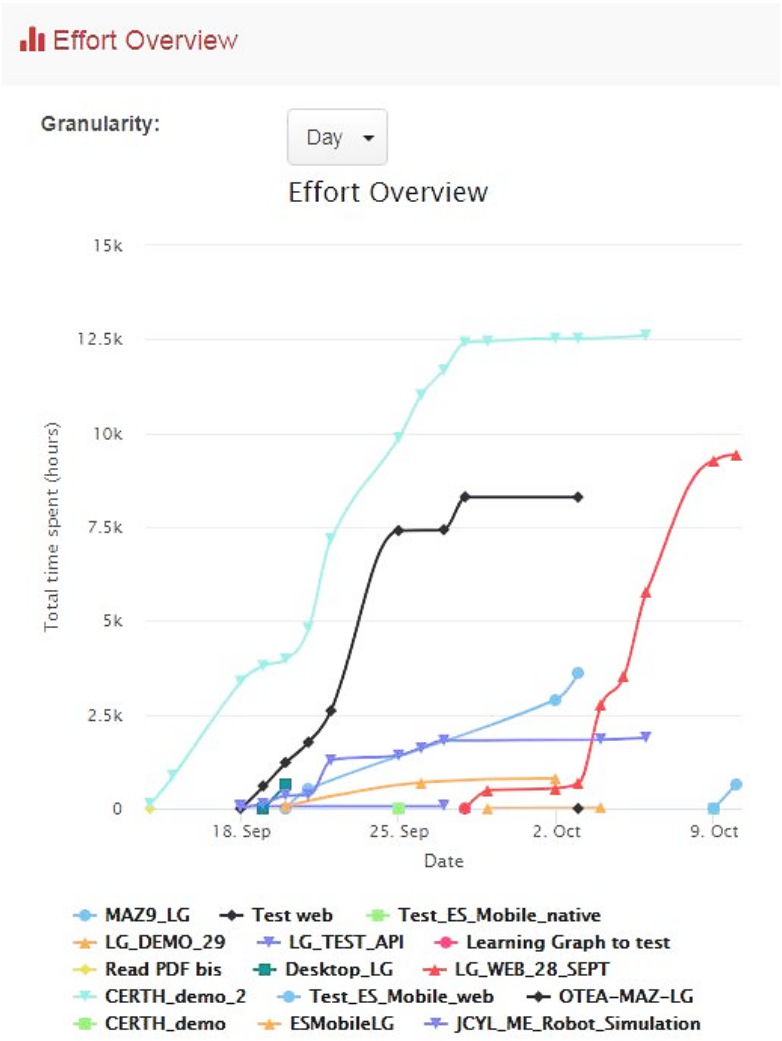

Figure 17 Effort overview for a classroom

As in learning progress overview case, it is also possible to select and check the effort taken in only one learning experience shown in Figure 18.

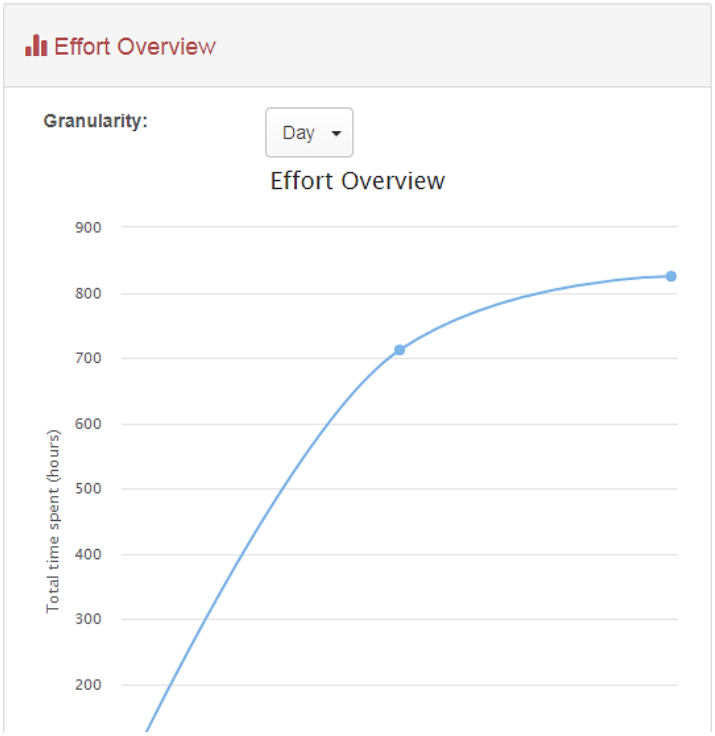

Figure 18 Effort overview for a selected learning experience (ESMobileLG)

Learning sessions overview by Learners

The learning sessions overview by learners is a pie chart (Figure 19) that compares the number of learners that have completed their learning sessions and the learners that have still running learning sessions. In the general case, where no learning experience is selected, the graph, taking into account all learners in the selected classroom, checks the state of these learners related learning sessions, and counts the number of learning sessions on each state: Finished and In progress. In this case, as it is by learner, if one learner has more than one session on the same state, it is only counted once.

Figure 19 Learning Sessions overview by learners for a classroom

By clicking on the details button  the tutor can view the details per learners, listing which learners are in progress and which have finished.

As on earlier graphs, here there is also the option of selecting a specific learning experience. At this case, the graph shows the same information, restricted to the learners taking part on selected learning experience.

Learning sessions overview by Sessions

Similar to the previous pie chart, the Learning sessions overview by sessions shows a comparison between how many sessions are in progress and how many completed within the classroom (Figure 20). In the general case, where no learning experience is selected, the graph, taking into account all learners in the selected classroom, checks the state of these learners related learning sessions, and counts the number of learning sessions on each state; Finished and In progress. In this case, as it is by sessions, if one learner has more than one session on the same state, all of them are counted.

Figure 20 Learning sessions overview by sessions for a classroom

If a specific learning experience is selected, the graph shows the same information, restricted to the learners taking part on selected learning experience.

Average time overview

The average time overview visualisation shows in the form of a bar chart (Figure 21) the average time spent by the learners in the classroom on each learning experience. In this way the tutor can have an insight on which are the learning experiences that take more time to complete by the learners. In the general case, where no specific learning experience is selected, the graph considers all learning experiences of all learners in the selected classroom. It calculates the average time, considering the time between the created date and last modified date for each learning experience.

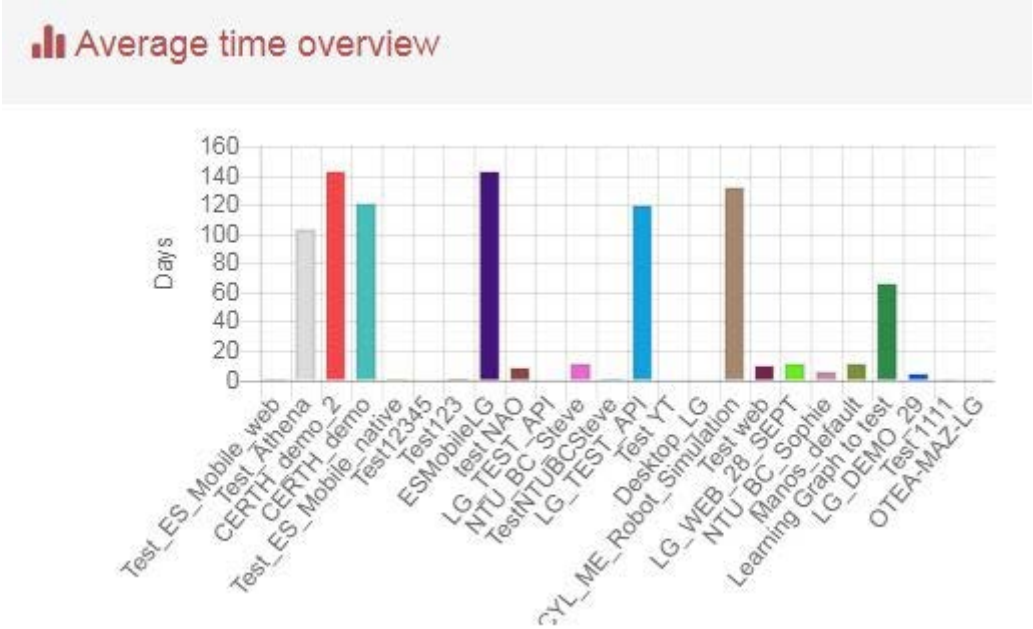

Figure 21 Average time overview for a classroom

If a specific learning experience is selected, the graph shows the same information, restricted to the selected learning experience (Figure 22).

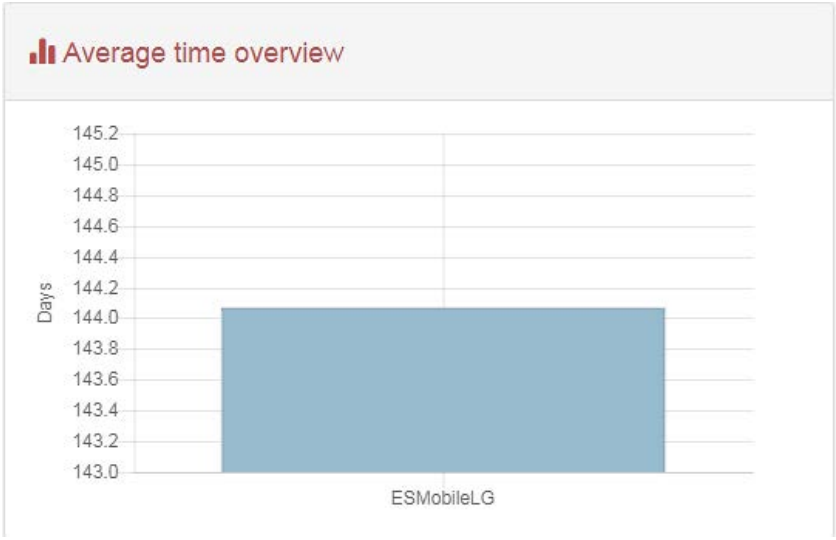

Figure 22 Average time overview for a learning experience

5.2.1.2 Learner overview

The tutor first needs to select the learner from the classroom to access the learner´s dashboard. When the learner is selected, the learner's dashboard is prepopulated. The dashboard can be viewed for all the learning experiences of the learner or for a specific learning experience.

Learning progress overview

The Learning Progress overview visualisation shows at a glance the average progress of the learner. It displays a line chart (Figure 23) with the average uptake on the presented learning contents of the learner per learning experience. The idea is similar as in "classroom overview" case, using the average weight of all the vertices of all learning experiences, but in this case, only for the selected learner: In the general case, where a learner is selected but no specific learning experience is selected, the graph shows the average weight of all the vertices in a learning experience, for all learning experiences where the selected learner takes part. This way, it shows how the knowledge of this learner is progressing in time, for all its learning experiences. The time granularity can be changed in order to view information per hour, day or month.

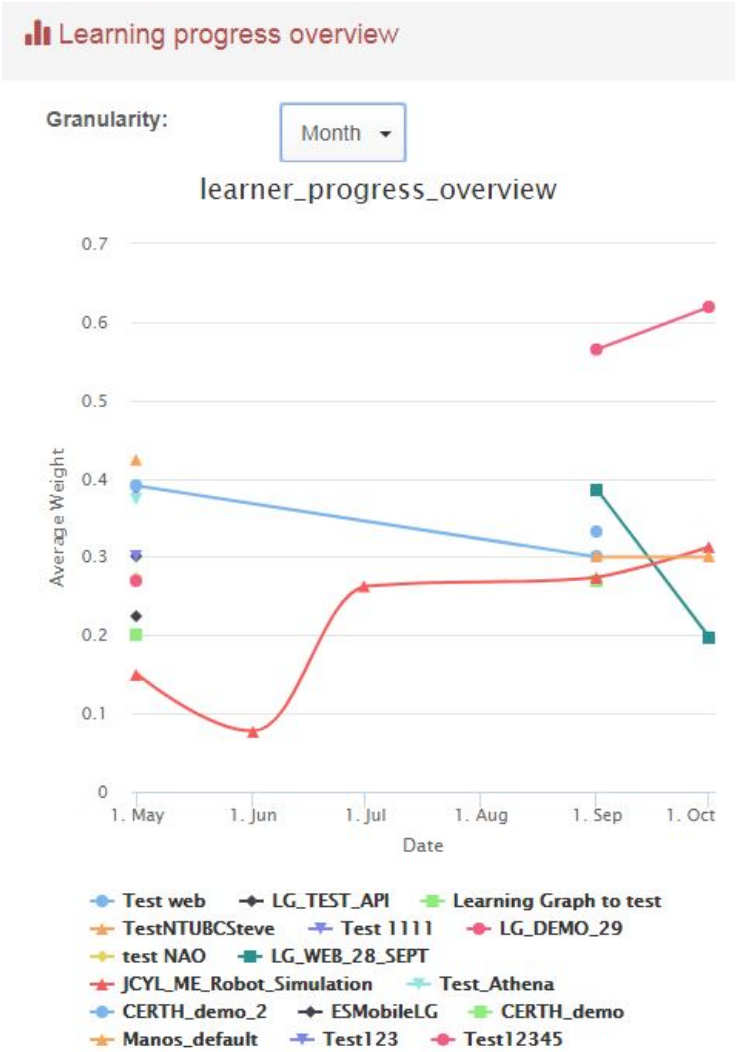

Figure 23 Learning progress overview for learner

Also, here there's the option to select a specific learning experience, and check only the progress of a learner on a concrete learning experience (Figure 24).

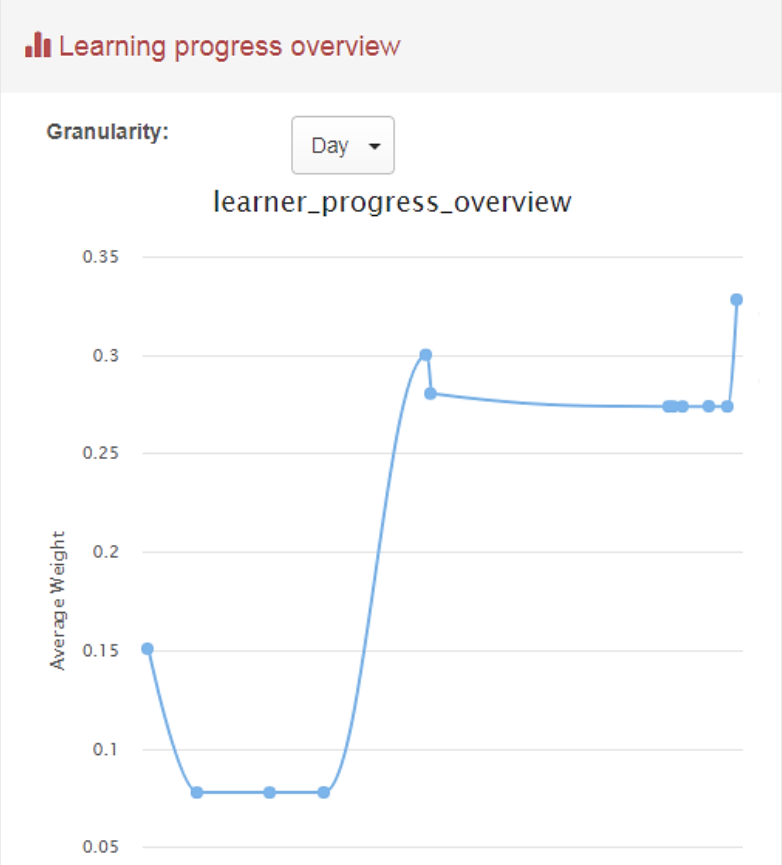

Figure 24 Learning progress overview for learner for a specific learning experience (JCYL_ME_Robot_Simulation)

Learning Experiences of learner

In this section the list of learning experiences is shown in more detail in the form of a table along with the start, end date and status (Figure 25). When a learning experience is selected, a timeline with the list of all learning sessions is presented with specific session information, such as the status, tutor, location, start and end date, platform agent as shown in Figure 21. When clicking on the session, a pop-up window displays more details on the session, as well as access to the personalised learning graph (Figure 26) where one can view the status of the graph with the SLAs and vertices weight as shown in Figure 22.

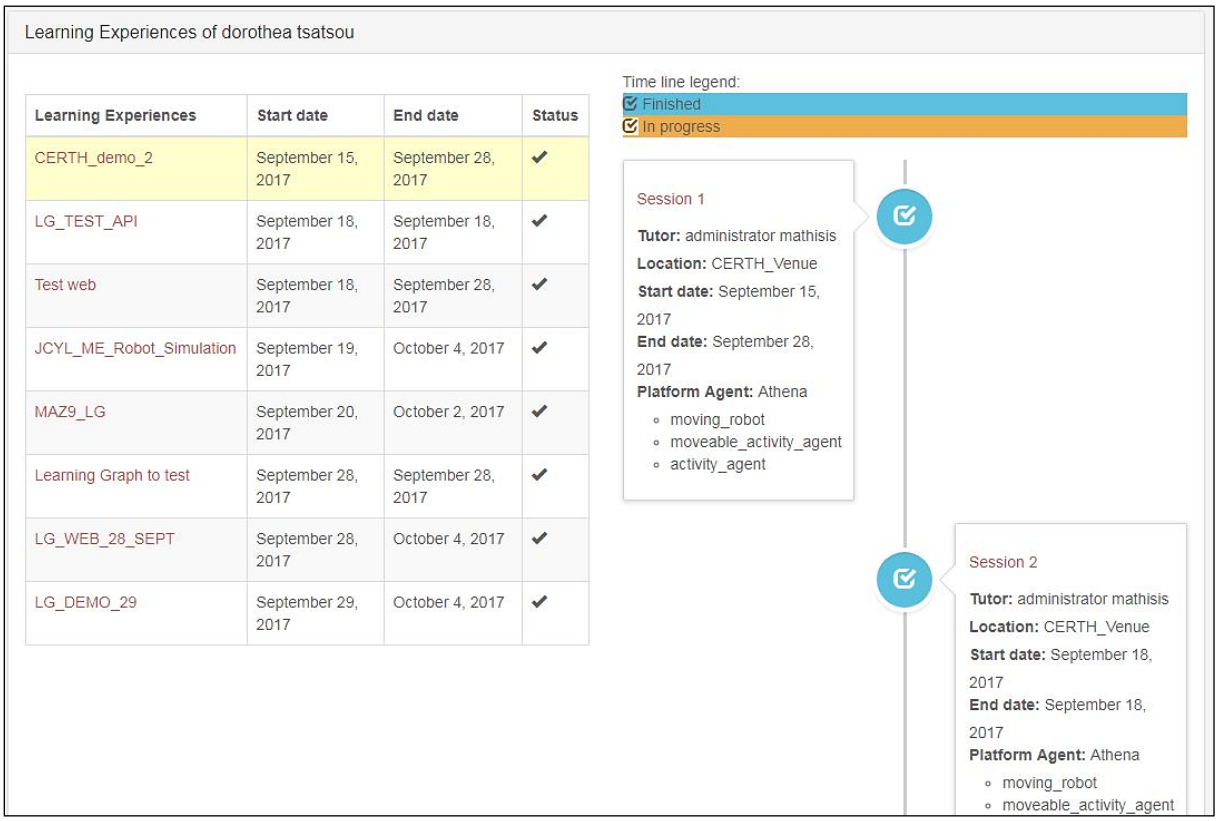

Figure 25 Learning experiences details for a learner

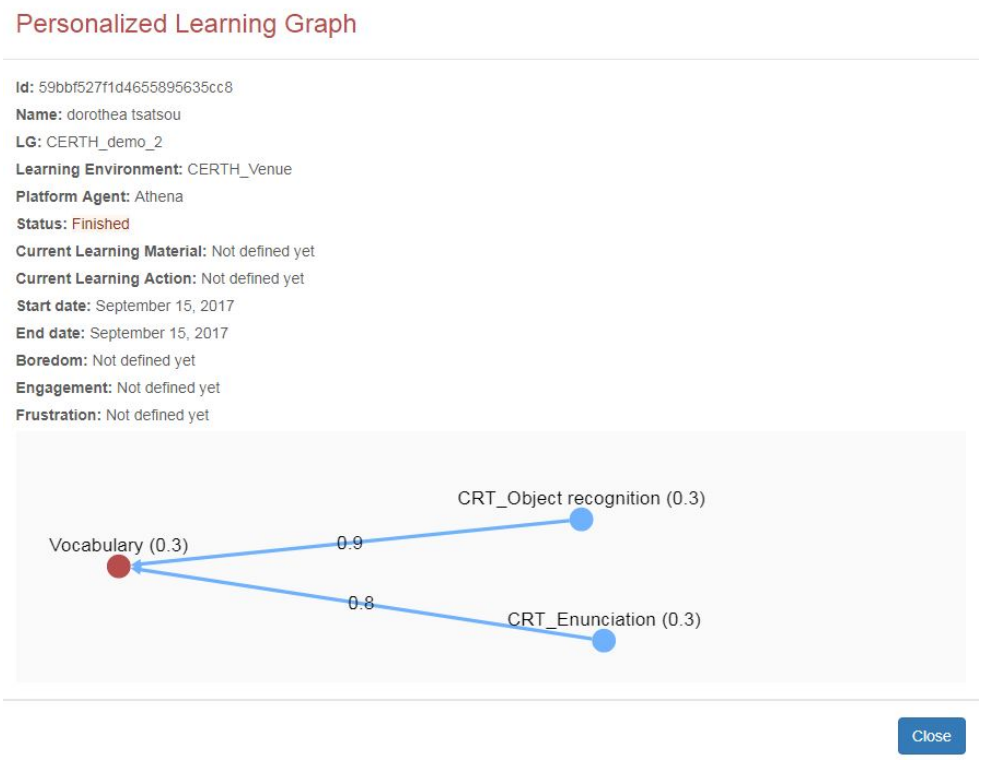

Figure 26 Learner´s personalised learning graph status

5.2.2 Caregiver dashboard

Users of MaTHiSiS with role of a caregiver, may only access the Learner overview section of the Analytics Dashboard. A dropdown appears with the list of learners that the caregiver is assigned. When a learner is selected, the learner overview dashboard is prepopulated as described in 5.2.1.2.

5.2.3 Independent learner dashboard

Independent learners have also access to their own learner's dashboard. When accessing it, the dashboard is prepopulated with the learner´s information as described in 5.2.1.2.

5.3 Plans for the next version

In the following months the Learning Analytics of MaTHiSiS will be further enriched with insights such as:

- The affective states of the learner
- The learner performance, coming from the implementation of the performance calculation as described in section 5.
- Learner collaboration, derived from ongoing work in WP6.
- Devices usage

Apart from the above mentioned plans, pilot users feedback is expected from their usage during the assisted pilots, which will be taken under consideration for further improvements. This feedback will hopefully enrich also the learning perspective of the Learning Analytics dashboard and provide useful information to the tutor and learners or their parents/caregivers.

6. Conclusion

In D4.5, the initial version of the affect and performance tracking algorithms are delivered through tracking learner sensorial data (facial expressions, eye gaze, body pose, voice) and by tracking learner responses to the learning material. A data flow diagram has been provided which shows the relationships between the different WP4 “Affective and Natural Interaction Instruments” tasks and the data interfaces that illustrate the flow of data in and out of each system component.

Specific interaction behaviour data (leaner’s response to questions / quiz challenges and response times) has been used as inputs to an algorithm that produces an objective value for learner performance. The learner’s sensory data was used to model the leaner’s emotional affect state as an outcome using robust machine learning algorithms. The affect detection and tracking methodology has been successfully integrated into the system and is the key component guiding the system adaptation to the learner. A summative emotional view of the current affect state of the learner has been presented as either ‘Bored’, ‘Frustrated’ or ‘Engaged’. This information will be used alongside learner performance to generate the forth affective state of ‘Flow’ in D4.6, where a fuzzy inference table for the values of affect state, performance and historic performance will produce recommendations in the adaptation (the difficulty or change) of the learning materials.

Algorithms for learner performance have been developed using proven methods for learner academic assessment. Elaborate learning analytics have been developed as part of T4.3 which will provide metrics of learner engagement at the level of subject, classroom and school scopes which is stored in the cloud. This would provide individual temporal and group based information for tutors and guide the leaner experience. These algorithms have been coded in part of functions that work as fulfilment of the learning analytics module in the MaTHiSiS system. The developed function has a robust method, which delivers a continuous flow of learner analytics logs. The database structure for this data has been described.

Finally, an initial version of multi-modal learner affect state detection and performance tracking methodology has been described and the code implementing it for the MaTHiSiS system has been developed and delivered in time for the Assisted Pilots.

7. Bibliography

[1] P. Ekman, “Facial expression and emotion,” American psychologist, vol. 48, no. 4, pp. 384– 92, 1993.

[2] G. M. Nagi, R. Wirza, F. Khalid, and M. Taufik, “Region-based facial expression recognition in still images,” Journal of Information Processing Systems, vol. 9, no. 1, 2013.

[3] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, C¸ . G¨ ulc¸ehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, R. C. Ferrari et al. , “Combining modality specific deep neural networks for emotion recognition in video,” in Proc. of the 15th ACM on Int. Conf. On Multimodal interaction, 2013, pp. 543–550.

[4] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The extended cohn-anade dataset (ck+): A complete dataset for action unit and emotion-specified expression,” in IEEE Conf. on Computer Vision and Pattern Recognition-Workshops , 2010, pp. 94– 101.

[5] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, “Coding facial expressions with gabor wavelets,” in Proc. of the IEEE Int. Conf. On Automatic Face Gesture Recognition and Workshops , 1998.

[6] T. Sim, S. Baker, and M. Bsat, “The cmu pose, illumination, and expression database,” IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) , vol. 25, no. 12, pp. 1615–1618, 2003.

[7] M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat, “Webbased database for facial expression analysis,” in Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME) , 2005, pp. 317–321. 1

[8] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, “Acted facial expres- sions in the wild database,” Australian National University, Canberra, Australia, Technical Report TR-CS-11 , vol. 2, 2011. 1

[9] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, “Coding facial expressions with gabor wavelets,” in Proc. of the IEEE Int. Conf. On Automatic Face and Gesture Recognition , 1998, pp. 200–205.

[10] C. Shan, S. Gong, and P. W. McOwan, “Facial expression recognition based on local binary patterns: A comprehensive study,” Image and Vision Computing , vol. 27, no. 6, pp. 803–816, 2009.

[11] S. Berretti, A. Del Bimbo, P. Pala, B. B. Amor, and M. Daoudi, “A set of selected sift features for 3d facial expression recognition,” in Int. Conf. on Pattern Recognition ICPR) , 2010, pp. 4125–4128.

[12] A. Asthana, J. Saragih, M. Wagner, and R. Goecke, “Evaluating AAM fitting methods for facial expression recognition,” in Proc. of the IEEE Int. Conf. on Affective Computing and Intelligent Interaction, ACII09 , 2009, pp. 598–605.

[13] D. Hamester, P. Barros, and S. Wermter, “Face expression recognition with a 2-channel convolutional neural network,” in Proc. of the Int. Joint Conf. on Neural Networks (IJCNN) , 2015.

[14] M. Liu, R. Wang, S. Li, S. Shan, Z. Huang, and X. Chen, “Combining multiple kernel methods on riemannian manifold for emotion recognition in the wild,” in Proc. of the 16th Int. Conf. On Multimodal Interaction , 2014, pp. 494–501.

[15] N. Srivastava and R. R. Salakhutdinov, “Multimodal learning with deep boltzmann machines,” in Advances in Neural Information Processing Systems , 2012, pp. 2222–2230.

[16] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in Proc. of the British Machine Vision Conference (BMVC) , 2015, pp. 1– 2.

[17]A. Batliner, S. Steidl, B. Schuller, D. Seppi, K. Laskowski, T. Vogt, L. Devillers, N. Vidrascu, L. K. Amir, and V. Aharonson, "Combining efforts for improving automatic classification of emotional user states," in Proceedings of IS-LTC , 2006, pp. 240–245.

[18]M. Soleymani, M. Pantic, and T. Pun, "Multimodal emotion recognition in response to videos," IEEE Transactions on Affective Computing , vol. 3, no. 2, pp. 211–223, 2012.

[19]F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in Proc. of the Int. Conf. On Computer Vision and Pattern Recognition (CVPR), 2006.

[20]R. Kullback, S.andLeibler, "On information and sufficiency," Annals of Mathematical Statistics, vol. 22, no. 1, pp. 79–86, 1951. 2, 6

[21]Z. Meng, S. Han, M. Chen, and Y. Tong, "Feature level fusion for bimodal facial action unit recognition," in IEEE Int. Symposium on Multimedia (ISM) , 2015.

[22]B. Jiang, B. Martinez, M. F. Valstar, and M. Pantic, "Decision level fusion of domain specific regions for facial action recognition," in Proc. of the IEEE Int. Conf. on Pattern Recognition (ICPR) , 2014.

[23]A. Dhall, R. Goecke, J. Joshi, M. Wagner, and T. Gedeon, "Emotion recognition in the wild challenge 2013," in ACM Conf. on Multimodal Interaction (ICMI'13) , 2013.

[24]Diginext: D2.4 – Full system architecture, 2016.

[25]University of Maastricht: D4.3 – Affect Understanding in MaTHiSiS, 2017.

[26]X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) , 2013, pp. 532–539.

[27]K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in Proc. of the British Machine Vision Conference (BMVC) , 2013.

[28]O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman, "A compact and discriminative face rack descriptor," in Proc. of the IEEE Conf. On Computer Vision and Pattern Recognition (CVPR) , 2014, pp. 1693– 1700.

[29]A. Saeed, A. Al-Hamadi, R. Niese, and M. Elzobi, "Effective geometric features for human emotion recognition," in IEEE 11th Int. Conf. On Signal Processing (ICSP) , vol. 1, 2012, pp. 623–627.

[30]H. Kaya, F. G˙ urpinar, S. Afshar, and A. A. Salah, "Contrasting and combining least squares based learners for emotion recognition in the wild," in Proc. of the Int. Conf. on Multimodal Interaction, 2015, pp. 459–466.

[31]F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in Proc. of the ACM Multimedia (MM) , 2013, pp. 835–838.

[32]B. Schuller, M. Valstar, F. Eyben, G. McKeown, R. Cowie, and M. Pan- tic, "Avec 2011–the first international audio/visual emotion challenge," in Int. Conf. on Affective Computing and Intelligent Interaction , 2011, pp. 415–424.

[33]J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," International Journal of Computer Vision (IJCV) , vol. 105, no. 3, pp. 222–245, 2013.

[34]D. Johnson and S. Sinanovic, "Symmetrizing the kullback- leibler dis- tance," IEEE Trans. on Information Theory , 2000.

[35]K.-C. Huang, H.-Y. S. Lin, J.-C. Chan, and Y.-H. Kuo, "Learn- ing collaborative decision-making parameters for multimodal emotion recognition," in IEEE Int. Conf. on Multimedia and Expo (ICME) , 2013, pp. 1–6.

[36]Centre for Research and Technology Hellas: D4.1 – MaTHiSiS sensorial component, 2016.

[37]T. Gehrig and H. K. Ekenel, "Why is facial expression analysis in the wild challenging?" in Proc. of the 2013 on Emotion recognition in the wild Challenge and Workshop , 2013, pp. 9–16.

[38]Nottingham Trent University: D6.1 – Adaptation and Personalization principles based on MaTHiSiS findings, 2016.

[39]Hoxby, C. M. (2002), The power of peers, *Education Next*, *2*(2), 57-63.

[40]Cain, A., & Burris, M. (1999). *Investigation of the use of mobile phones while driving*, http://www.cutr.eng.usf.edu/its/mobile_phone_text.htm, retrieved January 15, 2000.

[41]Greller, W., &Drachsler, H. (2012). Translating Learning into Numbers: A Generic Framework for Learning Analytics. Educational Technology & Society, 15 (3), 42–57

[42] Highcharts, (2017), Highcharts, https://www.highcharts.com/, retrieved October 11, 2017.

[43]MongoDB, (2017), Aggregation, https://docs.mongodb.com/manual/aggregation/, retrieved October 11, 2017.