



Real-Time Multimodal Feedback with the CPR Tutor

Daniele Di Mitri^{1(✉)}, Jan Schneider², Kevin Trebing³, Sasa Sopka⁴,
Marcus Specht⁵, and Hendrik Drachsler^{1,2}

¹ Open University of The Netherlands, Valkenburgerweg 177,
6419 AT Heerlen, The Netherlands

daniele.dimitri@ou.nl

² DIPF - Leibniz Institute for Research and Information in Education,
Frankfurt, Germany

³ Maastricht University, Maastricht, The Netherlands

⁴ AIXTRA - RWTH Aachen University Hospital, Aachen, Germany

⁵ Delft University of Technology, Delft, The Netherlands

Abstract. We developed the CPR Tutor, a real-time multimodal feedback system for cardiopulmonary resuscitation (CPR) training. The CPR Tutor detects mistakes using recurrent neural networks for real-time time-series classification. From a multimodal data stream consisting of kinematic and electromyographic data, the CPR Tutor system automatically detects the chest compressions, which are then classified and assessed according to five performance indicators. Based on this assessment, the CPR Tutor provides audio feedback to correct the most critical mistakes and improve the CPR performance. To test the validity of the CPR Tutor, we first collected the data corpus from 10 experts used for model training. Hence, to test the impact of the feedback functionality, we ran a user study involving 10 participants. The CPR Tutor pushes forward the current state of the art of real-time multimodal tutors by providing: 1) an architecture design, 2) a methodological approach to design multimodal feedback and 3) a field study on real-time feedback for CPR training.

1 Introduction

In learning science, there is an increasing interest in collecting and integrating data from multiple modalities and devices with the aim of analysing learning behaviour [4, 15]. This phenomenon is witnessed by the rise of multimodal data experiments especially in the contexts of project-based learning [21], lab-based experimentation for skill acquisition [11], and simulations for mastering psychomotor skills [19]. Most of the existing studies using multimodal data for learning stand at the level of “data geology”, investigating whether multimodal data can provide evidence of the learning process. In some cases, machine learning models were trained with the collected data for classifying or predicting outcomes such as emotions or learning performance. At the same time, the

existing research that uses multimodal and multi-sensor systems for training different types of psychomotor skills features neither personalised nor adaptive feedback [18].

In this study, we aimed at overcoming this knowledge gap and by *exploring how multimodal data can be used to support psychomotor skill development by providing real-time feedback*. We followed a design-based research approach: the presented study is based on the insights of [8], in which we demonstrated that it is possible to detect common CPR mistakes regarding the quality of the chest compressions (CC) (CC-rate, CC-depth and CC-release). In [8], we have also shown that it is possible to extend the common mistake detection of commercial and validated training tools like the Laerdal ResusciAnne manikin with the CPR tutor. We were able to detect the correct locking of the arms while doing CPR and the correct use of the body weight when performing the CCs. The mistake detection models were obtained training multiple recurrent neural networks, using the multimodal data as input and the presence or absence of the CPR mistakes as output. This study extends the previous efforts by embedding the machine learning approaches for mistake detection with real-time feedback intervention.

2 Background

2.1 Multimodal Data for Learning

With the term “multimodal data”, we refer to the data sources derived from multimodal and multi-sensor interfaces that go beyond the typical mouse and keyboard interactions [16]. These data sources can be collected using wearable sensors, depth cameras or Internet of Things devices. Example of modalities relevant for modelling a learning task is learner’s motoric movements, physiological signals, contextual, environmental or activity-related information [7]. The exploration of these novel data sources inspired the Multimodal Learning Analytics (MMLA) research [15], whose common hypothesis is that combining data from multiple modalities allows obtaining a more accurate representation of the learning process and can provide valuable insights to the educational actors, informing them about the learning dynamics and supporting them to design more valuable feedback [4]. The contribution of multimodal data to learning is still a research topic under exploration. Researchers have found out that it can better predict learning performance during desktop-based game playing [11]. The MMLA approach is also thought to be useful for modelling ill-structured learning tasks [5]. Recent MMLA prototypes have been developed for modelling classroom interactions [1] or for estimating success in group collaboration [21]. Multimodal data were also employed for modelling psychomotor tasks and physical learning activities that require complex body coordination [14]. Santos et al. reviewed existing studies using sensor-based applications in diverse psychomotor disciplines for training specific movements in different sports and martial arts [19]. Limbu et al. reviewed existing studies that modelled the experts to train apprentices using recorded expert performance [13].

2.2 Multimodal Intelligent Tutors

We are interested in the application of multimodal data for providing automatic and real-time feedback. This aim is pursued by the Intelligent Tutoring Systems (ITSs) research. Historically ITSs have been designed for well-structured learning activities in which the task sequence is clearly defined, as well as the assessment criteria and the range of learning mistakes that ITS is able to detect. Related ITS research looked primarily at meta-cognitive aspects of learning, such as the detection of learners' emotional states (e.g. [3, 10]). Several ITSs of this kind are reviewed in a recent literature review [2]. Most of these studies employed a desktop-based system where the user-interaction takes place with mouse and keyboard. To find applications of ITSs beyond mouse and keyboard we need to look in the field of medical robotics and surgical simulations into systems like DaVinci. These robots allow aspiring surgeons to train standardised surgical skills in safe environments [22].

2.3 Cardiopulmonary Resuscitation (CPR)

In this study, we focus on one of the most frequently applied and well studied medical simulations: Cardiopulmonary Resuscitation. CPR is a lifesaving technique applied in many emergencies, including a heart attack, near drowning or in the case of stopped heartbeat or breathing. CPR is nowadays mandatory not only for healthcare professionals but also for several other professions, especially those more exposed to the general public. CPR training is an individual learning task with a highly standardised procedure consisting of a series of predefined steps and criteria to measure the quality of the performance. We refer to the European CPR Guidelines [17]. There exists a variety of commercial tools for supporting CPR training, which can track and assess the CPR execution. A very common training tool is the Laerdal ResusciAnne manikins. The ResusciAnne manikins provide only retrospective and non-real-time performance indicators such as CC-rate, CC-depth and CC-release. Other indicators are neglected and that creates a feedback gap for the learner and higher responsibility for the course instructors. Examples of these indicators are the use of the body weight or the locking of the arms while doing the CCs. So far, these mistakes need to be corrected by human instructors.

3 System Architecture of the CPR Tutor

The System Architecture of the CPR Tutor implements the five-step approach introduced by the *Multimodal Pipeline* [9], a framework for the collection, storing, annotation, processing and exploitation of data from multiple modalities. The System Architecture was optimised to the selected sensors and for the specific task of CPR training. The five steps, proposed by the *Multimodal Pipeline* are numbered in the graphical representation of the System Architecture in Fig. 1. The architecture also features three layers: 1) the Presentation Layer interfacing with the user (either the learner or the expert); 2) the Application

Layer, implementing the logic of the CPR Tutor; 3) the Data Layer, consisting of the data used by the CPR Tutor. In the CPR Tutor, we can distinguish two main phases which have two corresponding data-flows: 1) the *offline training* of the machine learning models and 2) the *real-time exploitation* in which the real-time feedback system is activated.

3.1 Data Collection

The first step corresponds to the collection of the data corpus. The main system component responsible for the data collection is the CPR Tutor, a C# application running on a Windows 10 computer. The CPR Tutor collects data from two main devices: 1) the Microsoft Kinect v2 depth camera and 2) the Myo electromyographic (EMG) armband. In the graphic user interface, the user of the CPR Tutor can ‘start’ and ‘stop’ the recording of the session. The CPR Tutor collects the data of the user in front of the camera wearing the Myo. The collected data consist of:

- the 3D kinematic data (x,y,z) of the body joints (excluding ankles and hips)
- the 2D video recording from the Kinect RGB camera,
- 8 EMG sensors values, 3D gyroscope and accelerometer of the Myo.

3.2 Data Storing

The CPR Tutor adopts the data storing logic of the *Multimodal Learning Hub* [20], a core component of the *Multimodal Pipeline*. As the sensor applications collect data at different frequencies, at the ‘start’ of the session, each sensor application is assigned to a *Recording Object* a data structure arbitrary number of *Frame Updates*. In the case of the CPR Tutor, there are two main streams coming from the Myo and the Kinect. The *Frame Updates* contain the relative timestamp starting from the moment the user presses the ‘start’ until the ‘stop’ of the session. Each *Frame Update* within the same *Recording Object* shares the same set of sensor attributes, in the case of the CPR Tutor, 8 attributes for Myo and 32 for Kinect, corresponding to the raw features that can be gathered from the public API of the devices. The video stream recording from the Kinect uses a special type of *Recording Object*, specific for video data. At the end of the session, when the user presses ‘stop’, the data gathered in memory in the *Recording Objects* and the *Annotation Object* is automatically serialised into the custom format introduced by the LearningHub: the *MLT Session* (Meaningful Learning Task). For the CPR Tutor, the custom data format consists of a zip folder containing: the Kinect and Myo sensor file, and the 2D video in MP4 format. Serialising the sessions is necessary for creating the data corpus for the offline training of the machine learning models.

3.3 Data Annotation

The annotation can be carried out by an expert retrospectively using the *Visual Inspection Tool* (VIT) [6]. In the VIT, the expert can load the *MLT Session*

files one by one to triangulate the video recording with the sensor data. The user can select and plot individual data attributes and inspect visually how they relate to a video recording. The VIT is also a tool for collecting expert annotations. In the case of CPR Tutor, the annotations were given as properties of every single CC. From the SimPad of the ResusciAnne manikin, we extracted the performance metrics of each recorded session. With a Python script, we processed the data from the SimPad in the form of a JSON annotation file, which we added to each recorded session using the VIT. This procedure allowed us to have the performance metrics of the ResusciAnne manikin as “ground truth” for the training the classifiers. As previously mentioned, the Simpad tracks the chest compression performance monitoring three indicators, the correct CC-rate, CC-release and CC-depth. By using the VIT, however, the expert can extend these indicators by adding manually custom annotations, in the form of attribute-value pairs. For this study, we use the target custom classes *armsLocked* and *bodyWeight* corresponding to two performance indicators, currently not tracked by the ResusciAnne manikins.

3.4 Data Processing

For data processing, we developed a Python script named SharpFlow¹. This component is used both for the offline training and validation of the mistake detection classifiers as well as for the real-time classification of the single CCs. In the training phase, the entire data corpus (*MLT Sessions* with their annotations) is loaded into memory and transformed into two Pandas data frames, one containing the sensor data the other one containing the annotations. As the sensor data came from devices with different sampling frequencies, the sensor data frame had a great number of missing values. To mitigate this problem, the data frame was resampled into a fixed number corresponding to the median length of each sample. We obtained, therefore, a 3D tensor of shape ($\#samples \times \#attributes \times \#intervals$). The dataset was divided in 85% for training and 15% for testing using random shuffling. A part of the training set (15%) was used as validation set. We also applied feature scaling using min-max normalisation with a range of -1 and 1. The scaling was fitted on the training set and applied on the validation and test sets. The model used for classification was a Long-Short Term Memory network [12] which is a special type of recurrent-neural network. Implementation was performed using PyTorch. The architecture of the model chosen was a sequence of two stacked LSTM layers followed by two dense layers:

- a first LSTM with input shape 17×52 ($\#intervals \times \#attributes$) and 128 hidden units;
- a second LSTM with 64 hidden units;
- a fully-connected layer with 32 units with a sigmoid activation function;
- a fully connected layer with 5 hidden units (number of target classes)
- a sigmoid activation.

¹ Code available on GitHub (<https://github.com/dimstudio/SharpFlow>).

All of our classes have a binary class, so we use a binary cross-entropy loss for optimisation and train for 30 epochs using an Adam optimiser with a learning rate of 0.01.

3.5 Real-Time Exploitation

The real-time data exploitation is the run-time behaviour of the System Architecture. This phase is a continuous loop of communication between the CPR Tutor, the SharpFlow application and the prompting of the feedback. It can be summarised in three phases 1) detection, 2) classification and 3) feedback.

1) Detection. For being able to assess a particular action and possibly detect if some mistake occurs, the CPR Tutor has to be certain that the learner has performed a CC and not something different. The approach chosen for action detection is a rule-based approach. While recording, the CC detector continuously checks the presence of CCs by monitoring the vertical movements of the shoulder joints from the Kinect data. These rules were calibrated manually so that the CC detector finds the beginning and the end of the CCs. At the end of each CC, the CPR Tutor pushes the entire data chunk to SharpFlow via a TCP client.

2) Classification. SharpFlow runs a TCP server implemented in Python which is continuously listening for incoming data chunks by the CPR Tutor. In case of a new chunk, SharpFlow checks if it has a correct data format and if it is not truncated. If so, it resamples the data chunks and feeds them into the min-max scaler loaded from memory, to make sure that also the new instance is normalised correctly. Once ready, the transformed data chunk is fed into the layered LSTMs also saved in memory. The results for each of the five target classes are serialised into a dictionary and sent back to the CPR Tutor where they are saved as annotations of the CC. SharpFlow takes on average 70 ms to classify one CC.

3) Feedback. Every time the CPR Tutor receives a classified CC, it computes a performance and an Error Rate (ER) for each target class. The performance is calculated with a moving average with a window of 10s, meaning it considers only the CCs performed in the previous 10s. The Error Rate is calculated as the inverse sum of the performance: $ER_j = 1 - \sum_{i=0}^n \frac{P_{i,j}}{n}$ where j is one of the five target classes, n is the number of CCs in one time window of 10s. Not all the mistakes in CPR are, however, equally important. For this reason, we handcrafted five feedback thresholds of activation in the form of five rules. If the ER is equal or greater than this threshold the feedback is fired, otherwise, the next rule is checked. The order chosen was the following: $ER_{armsLocked} \geq 5$, $ER_{bodyWeight} \geq 15$, $ER_{classRate} \geq 40$, $ER_{classRelease} \geq 50$, $ER_{classDepth} \geq 60$. Although every CC is assessed immediately after 0.5s we set the feedback frequency to 10s, to avoid overloading the user with too much feedback. The modality chosen for the feedback was sound, as we considered the auditory sense the least occupied channel while doing CPR. We created the following audio messages for the five target classes:

(1) *classRelease*: “release the compression”; (2) *classDepth*: “improve compression depth”; (3) *armsLocked*: “lock your arms”; (4) *bodyWeight*: “use your body weight”; (5) *classRate*: *metronome sound at 110 bpm*.

4 Method

In light of the research gap on providing real-time feedback from multimodal systems, we formulated the following research hypothesis which guided our scientific investigation.

H1: The proposed architecture allows the provision of real-time feedback for CPR training.

H2: The real-time feedback of the CPR Tutor has a positive impact on the considered CPR performance indicators.

4.1 Study Design

To test H1, we developed the CPR tutor with a real-time feedback component based on insights from our design-based research cycle. We planned a quantitative intervention study in collaboration with a major European University Hospital. The study took place in two phases: 1) Expert data collection involving a group of 10 expert participants, in which the data corpus was collected; 2) a Feedback intervention study involving a new group of 10 participants. A snapshot of the study setup for both phases is shown in Fig. 2. All participants in the study were asked to sign an informed consent letter detailing all the details of the experiment as well as the treatment of the collected data in accordance with the new European General Data Protection Regulation (2016/679 EU GDPR).

4.2 Phase 1 - Expert Data Collection

The expert group counted 10 participants (M: 4, F: 6) having an average of 5.3 previous CPR courses per person. We asked the experts to perform 4 sessions of 1 min duration. Two of these sessions, they had to perform correct CPR, while the reminder two sessions they had to perform incorrect executions not locking their arms and not using their body weight. In fact, from the previous study [8] we noticed it was difficult to obtain the full span of mistakes the learners can perform. Asking the experts to mimic the mistakes was, thus, the most sensible option for obtaining a dataset with a balanced class distribution. We, therefore, collected around 400 CCs per participant. The 1 min duration was set to prevent that physical fatigue influenced the novice’s performance. Once the data collection was completed, we inspected each session individually using the *Visual Inspection Tool*. We annotated the CC detected by the CPR Tutor, by triangulating with the performance metrics from the ResusciAnne manikin. The *bodyWeight* and *armsLocked* were instead annotated manually by one component of the research team.

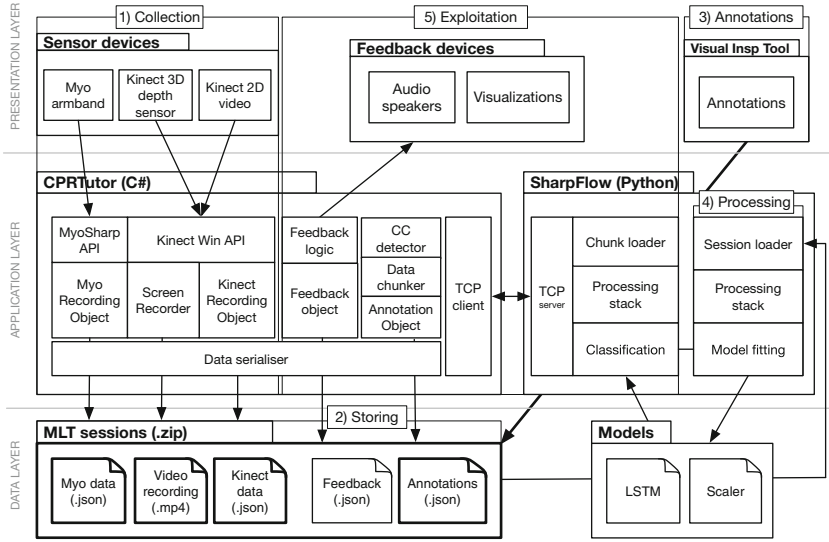


Fig. 1. The system architecture of the CPR tutor

4.3 Phase 2 - Feedback Intervention

The feedback intervention phase counted 10 participants (M: 5, F: 5) having an average of 2.3 previous CPR courses per person. Those were not absolute novices but recruited among the group of students that needed to renew their CPR certificate. The last CPR training for these participants was, therefore, older than one year. Each participant in the feedback intervention group performed 2 sessions of 1 min, one with feedback enabled and one without feedback.

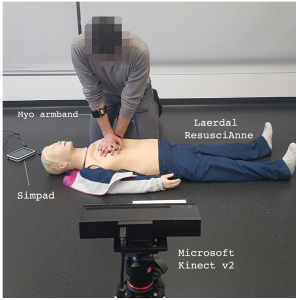
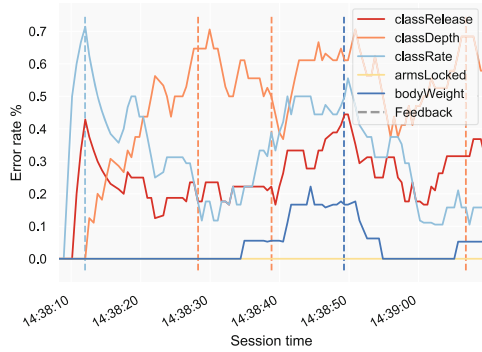
5 Results

The collected data corpus from the expert group consisted of 4803 CCs. Each CC was annotated with 5 classes. With the methodology described in Sect. 3.4, we obtained a tensor of shape (4803, 17, 52). As the distribution of the classes was too unbalanced, the dataset was downsampled to 3434 samples (-28.5%). In Table 1, we report the new distribution for each target class. In addition, we report the results of the LSTM training reporting for each target class the accuracy, precision, recall and F1-score. In the feedback group, we collected a dataset of 20 sessions from 10 participants with 2223 CCs detected by the CPR Tutor and classified automatically. The feedback function was enabled only in 10 out of 20 sessions. The feedback was fired a total of 16 times. In Table 2, we report the feedback frequency for each target class and the class distribution for each target class. We generated Error Rate plots for each individual session. In Fig. 3, we provide an example plot of a session having five feedback interventions (vertical dashed lines) matching the same colours of the target classes. Although

Table 1. Five target classes distribution and performance of corresponding LSTM models trained on the expert dataset.

Class	Class distribution	Accuracy	Precision	Recall	F1-score
classRelease	0: (1475, 42.9%), 1: (1959, 57.1%)	0.905	0.897	0.954	0.925
classDepth	0: (2221, 64.6%), 1: (1213, 35.4%)	0.954	0.955	0.953	0.954
classRate	0: (1457, 42.5%), 1: (1977, 57.5%)	0.901	0.815	0.819	0.817
armsLocked	0: (1337, 38.9%), 1: (2097, 61.1%)	0.981	0.975	1	0.987
bodyWeight	0: (1206, 35.1%), 1: (2228, 64.9%)	0.97	0.967	0.994	0.98

the Error Rates fluctuate heavily throughout each session, we noticed that nearly every time the feedback is fired the Error Rate for the targeted mistake is subject to a drop. We analysed, therefore, the effect of CPR Tutor feedback by focusing on the short-term changes in Error Rate for the mistakes targeted by the CPR Tutor. In Table 2, we report the average ERs 10s before and 10s after the audio feedback was fired. We report the average delta of these two values for each target class. For *classRelease*, *classDepth* and *classRate* we notice a decrease of the Error Rate, whereas for *armsLocked* and *bodyWeight* an average increase.

**Fig. 2.** Study design of the CPR tutor**Fig. 3.** Plot of the error rates for one session.

6 Discussion

In H1 we hypothesised that the proposed architecture for a real-time feedback is suitable for CPR training. With the System Architecture outlined in Sect. 3, we implemented a functional system which can be used both for the offline model training of the CPR mistakes as well as for the real-time multimodal data exploitation. The proposed architecture exhibited reactive performances, by classifying one CC in about 70 ms. The System Architecture proposed is the

Table 2. Average Error Rate for each target class 10 s before and 10 s after the audio feedback were fired.

Class	Class distribution	Freq. Feedback	ER before feedback	10 s feed-back	ER 10 s after feedback	Delta
classRelease	0: (475, 21.4%), 1: (1746, 78.6%)	2	46.60%		33.50%	-13.10%
classDepth	0: (704, 31.7%), 1: (1517, 68.3%)	5	59.80%		55.00%	-4.80%
classRate	0: (475, 21.4%), 1: (1746, 78.6%)	5	44.20%		34.5%	-9.70%
armsLocked	0: (3, 0.1%), 1: (2218, 99.9%)	1	0.6%		5.1%	4.50%
bodyWeight	0: (69, 3.1%), 1: (2152, 96.9%)	3	10.70%		12.90%	2.20%

first complete implementation of the *Multimodal Pipeline* [9] and it shows that it is possible to close the feedback loop with a real-time multimodal feedback.

In H2 we hypothesised that the CPR Tutor with its real-time feedback function can have a positive impact on the performance indicators considered. With a first intervention feedback study involving 10 participants we noticed that there is a short-term positive influence of the real-time feedback on the detected performance, witnessed by a decrease of Error Rate in the 10 s after the feedback was fired (Table 2). This effect is confirmed in three out of five target classes. The remaining two classes show opposite behaviours. In these two cases, the increase of Error Rate is smaller as compared to the former target classes. We suppose this behaviour is linked to the extreme class distribution of these two classes. In turn, this distribution can be due to the fact that the participants of the second group were not beginners and, therefore, not perform common mistakes such as not locking the arms or not using their body weight correctly. These observations cannot be generalised due to the small number of participants tested for the study.

7 Conclusions

We presented the design and the development of real-time feedback architecture for CPR Tutor. Building upon existing components, we developed an open-source data processing tool (SharpFlow) which implements a neural network architecture as well as a TCP server for real-time CCs classification. The architecture was employed in a first study aimed at expert data collection and offline training and the second study for real-time feedback intervention allowing us to prove our first hypothesis. Regarding H2, we collected observations that, while cannot be generalised, provide some indication that the feedback of the CPR tutor had a positive influence on the CPR performance on the target classes. To sum up, the architecture used for the CPR Tutor allowed for provision of real-time mul-

timodal feedback (H1) and the generated feedback seem to have a short-term positive influence on the CPR performance on the target classes considered.

References

1. Ahuja, K., et al.: EduSense: practical classroom sensing at scale. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 3, no. 3, pp. 1–26 (2019). <http://dl.acm.org/citation.cfm?doi=3361560.3351229>
2. Alqahtani, F., Ramzan, N.: Comparison and efficacy of synergistic intelligent tutoring systems with human physiological response. *Sensors* **19**(3), 460 (2019)
3. Arroyo, I., Cooper, D.G., Burleson, W., Woolf, B.P., Muldner, K., Christopherson, R.: Emotion sensors go to school. *Front. Artif. Intell. Appl.* **200**(1), 17–24 (2009)
4. Blikstein, P., Worsley, M.: Multimodal learning analytics and education data mining: using computational technologies to measure complex learning tasks. *J. Learn. Anal.* **3**(2), 220–238 (2016)
5. Cukurova, M., Kent, C., Luckin, R.: Artificial intelligence and multimodal data in the service of human decision-making: a case study in debate tutoring. *Br. J. Educ. Technol.* **50**, 3032–3046 (2019). <https://onlinelibrary.wiley.com/doi/abs/10.1111/bjet.12829>
6. Di Mitri, D., Schneider, J., Klemke, R., Specht, M., Drachslar, H.: Read between the lines: an annotation tool for multimodal data for learning. In: Proceedings of the 9th International Conference on Learning Analytics & Knowledge - LAK19, pp. 51–60. ACM, New York (2019). <http://dl.acm.org/citation.cfm?doi=3303772.3303776>
7. Di Mitri, D., Schneider, J., Specht, M., Drachslar, H.: From signals to knowledge: a conceptual model for multimodal learning analytics. *J. Comput. Assist. Learn.* **34**(4), 338–349 (2018). <http://doi.wiley.com/10.1111/jcal.12288>
8. Di Mitri, D., Schneider, J., Specht, M., Drachslar, H.: Detecting mistakes in CPR training with multimodal data and neural networks. *Sensors* **19**(14), 1–20 (2019)
9. Di Mitri, D., Schneider, J., Specht, M., Drachslar, H.: Multimodal pipeline : a generic approach for handling multimodal data for supporting learning. In: AIMA4EDU Workshop in IJCAI 2019 AI-based Multimodal Analytics for Understanding Human Learning in Real-world Educational Contexts, pp. 2–4 (2019)
10. D’Mello, S., et al.: AutoTutor detects and responds to learners affective and cognitive states. *IEEE Trans. Educ.* **48**(4), 612–618 (2008)
11. Giannakos, M.N., Sharma, K., Pappas, I.O., Kostakos, V., Velloso, E.: Multimodal data as a means to understand the learning experience. *Int. J. Info. Manag.* **48**(Feb), 108–119 (2019). <https://doi.org/10.1016/j.ijinfomgt.2019.02.003>
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997). <http://www7.informatik.tu-muenchen.de/hochreit%0Awww.idsia.ch/juergen>
13. Limbu, B.H., Jarodzka, H., Klemke, R., Specht, M.: Using sensors and augmented reality to train apprentices using recorded expert performance: a systematic literature review. *Educ. Res. Rev.* **25**(2017), 1–22 (2018). <https://doi.org/10.1016/j.edurev.2018.07.001>
14. Martinez-Maldonado, R., Echeverria, V., Santos, O.C., Dos Santos, A.D.P., Yacef, K.: Physical learning analytics: a multimodal perspective. In: ACM International Conference Proceeding Series, pp. 375–379 (2018)

15. Ochoa, X., Worsley, M.: Augmenting learning analytics with multimodal sensory data. *J. Learn. Anal.* **3**(2), 213–219 (2016). <http://learning-analytics.info/journals/index.php/JLA/article/view/5081>
16. Oviatt, S., Schuller, B., Cohen, P.R., Sonntag, D., Potamianos, G., Krüger, A.: *The Handbook of Multimodal-Multisensor Interfaces: Foundations, User Modeling, and Common Modality Combinations*, vol. 2, April 2018. <https://dl.acm.org/citation.cfm?id=3015783>
17. Perkins, G.D., et al.: European resuscitation council guidelines for resuscitation 2015: Section 2. Adult basic life support and automated external defibrillation. *Resuscitation* **95**, 81–99 (2015). <https://doi.org/10.1016/j.resuscitation.2015.07.015>
18. Santos, O.C.: Training the body: the potential of AIED to support personalized motor skills learning. *Int. J. Artif. Intell. Educ.* **26**(2), 730–755 (2016). <https://doi.org/10.1007/s40593-016-0103-2>
19. Santos, O.C.: Artificial intelligence in psychomotor learning: modeling human motion from inertial sensor data. *Int. J. Artif. Intell. Tools* **28**(04), 1940006 (2019)
20. Schneider, J., Di Mitri, D., Limbu, B., Drachsler, H.: Multimodal learning hub: a tool for capturing customizable multimodal learning experiences. In: Pammer-Schindler, V., Pérez-Sanagustín, M., Drachsler, H., Elferink, R., Scheffel, M. (eds.) *EC-TEL 2018*. LNCS, vol. 11082, pp. 45–58. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98572-5_4
21. Spikol, D., Ruffaldi, E., Dabisias, G., Cukurova, M.: Supervised machine learning in multimodal learning analytics for estimating success in project-based learning. *J. Comput. Assist. Learn.* **34**(4), 366–377 (2018)
22. Taylor, R.H., Menciassi, A., Fichtinger, G., Fiorini, P., Dario, P.: Medical robotics and computer-integrated surgery. In: Siciliano, B., Khatib, O. (eds.) *Springer Handbook of Robotics*, pp. 1657–1684. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32552-1_63