Visual and Memory-based HCI Obstacles: Behaviour-based Detection and User Interface Adaptations Analysis

Mazen Salous*[1], Felix Putze[1], Markus Ihrig[1] and Tanja Schultz[1]

Abstract— **Human Computer Interaction (HCI) performance can be impaired by several HCI obstacles. Cognitive adaptive systems should dynamically detect such obstacles and compensate them with suitable User Interface (UI) adaptation. In this paper, we discuss the detection of two main HCI obstacles: memory-based and visual obstacles. A sequential model based on Long-Short Term Memory (LSTM) is suggested for such a detection of HCI obstacles. UI adaptations for both types of obstacles are discussed and analyzed. We investigate the classification performance on data from a user study with 17 participants. Furthermore, we also investigate the influence of different adaptation mechanisms on performance and subjective assessment. Results show advantages of the proposed sequential LSTM model: on the one hand, the LSTM outperforms the baseline random guess and also a baseline static model LDA in the detection of visual obstacles with 70.6% as an average accuracy. On the other hand, the evaluation of HCI sessions impeded by obstacles but supported with different UI adaptations shows that LSTM results well match the subjective assessment as a plausible detector of behaviour changes.**

I. INTRODUCTION

Cognitive adaptive systems [16] aim at automatically adapting to different user groups and specific situations. For example, some users can handle a specific Human Computer Interaction (HCI) task smoothly, while others may suffer from difficulties, so-called *interaction obstacles*, during the same task. Humans employ several perceptual, cognitive, and motor capabilities while interacting with computers. Limitations on these capabilities can lead to a deteriorated HCI performance and this requires suitable User Interface (UI) adaptation as compensation, e.g. [5], [8]. While persistent interaction obstacles, like a deficiency in motor capabilities, can be handled by specific limitation-oriented systems, different types of interaction obstacles need different UI adaptations for different users: Compensating an interaction obstacle with a wrong or unnecessary adaptation can be harmful for the user performance and acceptance. This is especially true in the case of transient interaction obstacles, which only occur temporarily, e.g. memory-based interaction obstacle caused temporarily by an HCI secondary task workload. UI adaptations differ also regarding to the amount of changes applied in the UI: A light UI adaptation makes minor changes while a strong UI adaptation makes major changes in the UI.

One important example of a transient interaction obstacle is increased memory load, caused by a secondary task. It may lead to a user forgetting already presented information or getting lost in a navigation hierarchy of the UI. As an example for a persistent obstacle, statistics show that 8% of men and 0.5% of women with Northern European ancestry have the common form of red-green color blindness [9], thus, a lot of potential computer users may suffer from bad recognition of colored items in the UI. While such an obstacle can be tackled statically by enabling users to customize the color-scheme in the UI, an online detection and compensation of color blindness interaction obstacle would be with greater acceptance from all potential users.

In this paper, we discuss the detection and compensation of those very important interaction obstacles: secondary task in HCI as *memory-based interaction obstacle* and red-green color vision deficiency as *visual obstacle*. We used a tablet-based implementation of the well-known *matching pairs game* (memory game) as an exemplary HCI application since it not only relies on memory but also represents a complex interaction task, which requires users' recognition ability of the visual items (cards show only colors, Figures 1 and 2). In a single-Player mode, the user should find all the pairs with as few turns as possible. Users' actions (the selected cards) while approaching this task continuously indicate to their working memory status and playing behavior. Player behaviour is encapsulated in the recorded behavioral data (sequences of user actions, i.e. the selected cards). We train classifiers to detect from that behavioral data whether the tested HCI application (matching pairs game) is impeded by an interaction obstacle (memory-based or visual obstacle) or not.

Our ultimate goal is to develop a fully automatic cognitive adaptive system that detects such interaction obstacles, examines possible UI adaptations and applies one or more UI adaptations that best fit the impeded user to improve the HCI performance. In previous works, the behaviour-based detection of memory-based interaction obstacles has been discussed and evaluated using Long Short-Term Memory networks [4]. In this paper, we extend this model to also cover the detection of visual obstacles. Furthermore, we look at behaviour changes and subjective evaluation when applying different UI adaptations to compensate for such obstacles. For this purpose, we analyze HCI sessions under different conditions: HCI sessions without obstacles (neither

*This work has been done within the project DINCO "Detection of Interaction Competencies and Obstacles". We thank the German Research Foundation (DFG) for funding DINCO project under the reference number PU 613/1-1.

We thank Rafael Miranda for implementing the first version of the matching-pairs application with UI adaptations.

[1] Authors work in the Cognitive Systems Lab (CSL) at the University of Bremen, Germany. firstname.lastname@uni-bremen.de

memory-based nor visual), sessions impeded by an obstacle, and sessions impeded by an obstacle but supported with a UI adaption. Besides a statistical analysis, we also investigate how such supported sessions are classified by the obstacle detector. While statistics can show for example that a specific adaptation improves the user's performance during HCI, a classifier evaluation responds to more subtle behavior changes.
The main contributions of this paper are: 1) Cognitive user simulation of plausible HCI sessions with and without different HCI obstacles. 2) Employing a sequential model that exploits potential temporal dependencies in behavioural data to detect different HCI obstacles. 3) Investigation of different UI adaptations from multiple perspectives.

II. Related Work

We discuss related works in two main topics: memory-based HCI obstacles and color vision HCI Obstacles.

A. *Memory-based HCI Obstacles*

Several approaches discussed memory obstacle in HCI. While many approaches e.g. [1], [2], [3] detect memory-based obstacles (simulated as secondary task) from observed physiological sensory data (e.g. electroencephalography EEG), other approaches e.g. Putze et al. [4] introduced an LSTM model to detect memory-based obstacle in HCI from pure behavioral data, i.e. from encoded user actions and without any noisy sensory data. Moreover, for memory-based HCI obstacles and UI adaptation, Sguerra et al. [5] investigated the online UI adaptation based on real-time tracking of human working memory. Sguerra modeled the human memory based on the Moran process [6] which is typically used to model the dynamics of finite populations in biology. For human memory, the stochastic model maintains ”quanta” numbers (weights) for each stored item. As an HCI, Sguerra used also a matching pairs game but without an explicit secondary task as memory-based obstacle. Instead, his stochastic model tracks the user performance and releases an adaptation signal when the performance deteriorates to a value less than a given application-based parameter. However, Sguerra's approach is an explicit performance-based tracking model, which does not consider potential temporal dependencies in behavioral data (user actions). Thus, in case of a secondary task memory obstacle, this model can only detect an impaired/enhanced performance (similar to our baseline static model LDA, see Section VI-C) but it cannot detect behavioural changes when applying that UI adaptation while the explicit secondary task is still existing. This is a contribution of our introduced sequential model LSTM, see the evaluation Section VI-C.

B. *Color Vision HCI Obstacles*

An early work presented by Jefferson et al. [7] introduced an interactive interface for users with color vision deficiency. Users can customize the colours of any region of the screen. Thus, it is a specific interface for such a user group and it permits the user to capture, clear, copy, save the image, and view the correction control window. In other words, there is no detection of color vision deficiency obstacle in this interface. Due to its relative high population and its strong impairment expected in HCI performance, there exist works that aim at online detecting color vision deficiency visual obstacle and online compensating with corresponding UI adaptation. For example, Khan et al. [8] discussed the online detection of such an obstacle by utilizing Ishihara plates test (Typical color vision test automated in mobile app). After that test, the most appropriate color scheme for that specific user is chosen. The limitation of such a UI that it is a specific color test cannot be smoothly integrated in HCI applications. Although Khan's UI adaptation has an advantage over Jefferson's approach as the most appropriate color scheme is chosen automatically, it can be further optimized by utilizing another cognitive ability e.g. by voice instructions. To the best of our knowledge, there is no approach in the literature that detects color vision deficiency from user behavioral data and adapts the UI by utilizing an additional user cognitive ability (see our introduced approach in Section III).

III. Data Collection and Protocol Design

In this section, we describe the experiments designed to collect users' behavioral data during HCI (matching pairs game) under different conditions (different matching pairs game variants). The matching pairs game variants are implemented for participants on an Android tablet. First, the participants were given a description for each game variant. Then, before each game variant started, each participant had a trial phase for that game variant to learn it. After each individual game, the participants were also asked to fill out a questionnaire about the played game variant, see Section IV for details about such questionnaires. In total, 17 participants (students: 13 male, 4 female, between 17 to 27 years old) completed our experiments. All participants gave their informed written consent. The data collection was approved by the ethics committee of the University of Bremen. The participants were asked to play the following variants of the matching pairs game, in randomized order, to collect their corresponding behavioral data (user actions, i.e. the selected cards).

- **NOOBS_NOADAPT: No Obstacle No Adaptation:** The cards of this game show well-distinguishable colors, See Figure 1.
- **MEMOBS_NOADAPT: Memory Obstacle No Adaptation:** The same game as above but impaired by a secondary task: For each revealed card, the participant will hear a random number between 1 and 9 from the tablet synthesized voice. The participant is asked to calculate the sum of all spoken numbers throughout the game. At the end of the game, the App prompts asking the participant for the final sum.
- **MEMOBS_STMEMADAPT: Memory Obstacle Strong Memory Adaptation:** The same memory-obstacle game, but the participants are supported with a `strong UI adaptation`, where all previously revealed cards

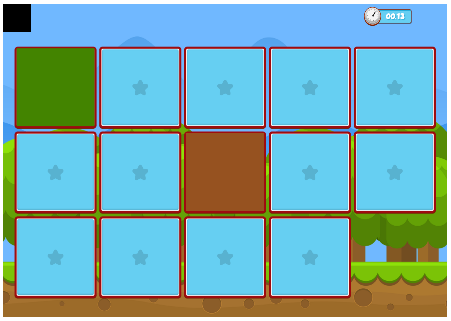

Fig. 1. Cards in standard game show well-distinguishable colors

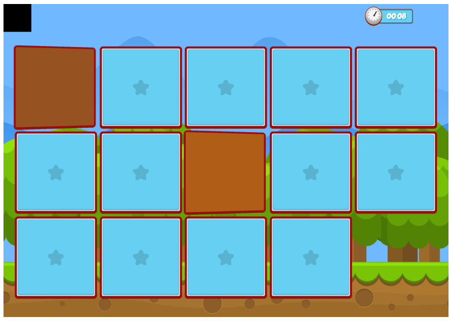

Fig. 2. Cards show red/green shaded colors as an emulation to red-green color vision deficiency

are re-revealed whenever a player selects non-matching cards.

- **MEMOBS_LIMEMADAPT: Memory Obstacle Light Memory Adaptation:** The participants of memory-obstacle game are supported with a `light UI adaptation`: Whenever a player reveals non-matching cards, only the last two cards revealed in the round before are re-revealed. Showing 4 cards in a very short time frame might allow the player to pick out a pair easier.
- **VISOBS_NOADAPT: Visual Obstacle No Adaptation:** In contrast to the first mentioned game variant (without obstacles), the cards of this game only show different shades of brown color to emulate red-green color vision deficiency as a visual obstacle game variant. See Figure 2.
- **VISOBS_VOICADAPT: Visual Obstacle Voice Adaptation:** The same visual-obstacle game, but the participants are supported with an additional channel assistance: *voice assistance*. While simulated red-green color vision deficiency cards make it difficult for players to distinguish the cards pairs, an identifier letter is spoken by the synthesized voice for each revealed card. Thus, seven different audio-identifier letters were needed to be spoken to help the user distinguishing between the seven pairs of cards in this 14-cards game (a,c,j,q,x,v and l were chosen to be easily distinguishable in the German pronunciation). With such spoken identifiers, the participant can compensate the red-green color vision deficiency by mapping the spoken letters to their corresponding positions.

IV. Statistical Analysis

In this section, we analyze the results of the participants' Likert-scale questionnaires and game logs. Following Vieira [17], we apply t-Tests for comparison of questionnaire items. Thus, we carried out paired t-tests between different game variants to test for significant differences. To correct for multiple comparisons in the questionnaire evaluation, we apply Bonferroni-correction to change the significance level from $p < 0.05$ to $p < 0.05/7 = 0.007$.

TABLE I

The main game variants (without adaptations) in comparison: average and standard deviation for questionnaire responses and game metrics.

Question	**noObs.**	**memObs.**	**visObs.**
Mental demand	3.8 (1.6)	6.4 (0.7)	4.2 (1.4)
Assessment of speed	6.0 (1.0)	3.1 (1.8)	4.7 (1.2)
Card memorability	6.1 (0.8)	3.4 (1.6)	3.9 (1.7)
Time needed [s]	41.8 (15.9)	134.4 (49.5)	60.4 (22.9)
Turns needed [#]	13.8 (2.8)	16.6 (3.1)	16.3 (2.9)
Errors made [#]	2.3 (2.3)	5 (3.1)	4.5 (3.1)

A. Questionnaire Analysis

A specific questionnaire for each game variant mentioned above was given to the participant when she or he finished that game variant. The important statements to be examined were "The game was mentally demanding", "I think I played through the game quickly" and "I could memorize the position of the cards well" as well as "I could sum the numbers together without problems" and "The assistance was helpful" for different game variants. For each statement, participants could assign a score between 1 and 7, with 7 being the maximum approval (Likert scale). Table I (upper half) shows that according to mental demand, assessment of speed and card memorability questions, the players found the standard game to be the easiest, the visual obstacle game to be slightly harder and the memory-based obstacle game to be significantly harder (with $p = 0.25$, $p = 0.001$, $p < 0.001$ for the NOOBS_NOADAPT vs. VISOBS_NOADAPT three comparisons respectively, and $p < 0.001$ for all the three comparisons of NOOBS_NOADAPT vs. MEMOBS_NOADAPT).

Table II shows comparisons between each game with obstacle and its assisted game variant, e.g. MEMOBS_NOADAPT vs. MEMOBS_LIMEMADAPT. When comparing the MEMOBS_NOADAPT to MEMOBS_LIMEMADAPT game variants, we found no significant differences between the players assessments of their performances, nor of their assessment of completing the arithmetic task more easily (p-values regarding the metrics order in the table: $p = 1$, $p = 0.27$, $p = 0.07$, $p = 1$). However, the MEMOBS_STMEMADAPT game variant shows a significant difference in players' performance assessment (mental demand and card memorability

TABLE II

UI Adaptations in comparison: average and standard deviation for questionnaire responses and game metrics.

Question	Without assistance	With assistance
memObs_noAdapt vs. memObs_LiMemAdapt		
Mental demand	6.4 (0.7)	6.4 (0.7)
Assessment of speed	3.1 (1.8)	3.7(1.7)
Card memorability	3.4 (1.6)	3.5(1.5)
Ability to sum	3.5 (1.6)	3.5 (1.5)
assistance helpfulness	-	5.1 (2.0)
Time needed	134.4 (49.5)	119 (40.4)
Turns needed	16.6 (3.1)	14.8 (2.3)
Errors made	5 (3.1)	3.2 (2.4)
memObs_noAdapt vs. memObs_StMemAdapt[1]		
Mental demand	6.1 (0.8)	5.5 (1.1)
Assessment of speed	4 (1.3)	3.8 (1.5)
Card memorability	3.7 (1.2)	4.6 (1.6)
Ability to sum	3.3 (1.2)	3.4 (1.3)
assistance helpfulness	-	5.4 (1.5)
Time needed	110 (22.9)	159 (41)
Turns needed	14.6 (1.9)	13.4 (1.6)
Errors made	5.4 (3.1)	2.9 (2.1)
visObs_noAdapt vs. visObs_voicAdapt		
Mental demand	4.2 (1.4)	3.7 (1.6)
Assessment of speed	4.7 (1.2)	5.9 (1.2)
Card memorability	3.9 (1.7)	6 (1.3)
Assistance helpfulness	-	6.7 (0.9)
Time needed	60.4 (22.9)	38.5 (14)
Turns needed	16.3 (2.9)	12.3 (3)
Errors made	4.5 (3.1)	1.4 (2.5)

questions).

With the visObs_noAdapt vs. visObs_voicAdapt game variants, however, the players found the corresponding voice-assistance to be very helpful, giving it the highest average score in that regard. They also felt they played faster ($p = 0.005$) and could memorize the cards more easily ($p < 0.001$). Only the question concerning mental workload found no significant difference ($p = 0.23$).

B. Game-log Analysis

The game logs recorded the first 20 turns and the total time needed. Additionally, game logs allow the calculation of error values. An error in this game is considered when a player turns a card whose partner has been already seen before, but fails to pick up the pair. Table I (lower half) shows results coincide with the players assessments discussed in the upper half of the table. Concretely, Table I shows that the noObs_noAdapt game has the best overall performance considering time needed, turns needed and errors made. The visObs_noAdapt game shows worse results and the memObs_noAdapt game the worst. The most significant differences are the time difference ($p < 0.001$) and errors ($p < 0.001$) between the noObs_noAdapt and memObs_noAdapt obstacle games, both being more than doubled in the latter.

Similarly, matching the questionnaire analysis in Section IV-A, Table II shows that the memObs_LiMemAdapt game came with no significant increases in performance of any of the three aforementioned metrics compared to the memObs_noAdapt game (p-values regarding the metrics order in the table: $p = 0.3$, $p = 0.06$, $p = 0.1$).

Table II also shows that the visObs_voicAdapt game reduces time ($p < 0.001$), turns ($p < 0.001$) and errors ($p < 0.001$) made compared to visObs_noAdapt game. Errors especially, were reduced by over 70% on average.

V. Classification Setup

The behavioural data episodes –recorded from different game variants e.g. noObs_noAdapt, memObs_noAdapt etc.– are classified using binary classification models into two labels: `No Obstacle` or `HCI Obstacle`. With the two different HCI obstacles (visual and memory-based obstacles), two different user behaviours are expected. Therefore, we prepared two separated classification models accordingly. That is, a binary classification model is trained to detect visual obstacles, and another model is trained to detect memory-based obstacles. While such classifiers can be clearly evaluated using HCI sessions with and without obstacles, the sessions supported with UI adaptations are also used to evaluate those classifiers to check how well the classifiers can detect behaviour changes caused by applying strong or light UI adaptation; With good obstacle compensation provided by adaptation, we expect the classifiers to classify the presented obstacle as ”no obstacle”.

While discriminant models –such as Linear Discriminant Analysis LDA and Support Vector Machine SVM– solely depend on manually, predefined features to discriminate data episodes, sequential models –such as Recurrent Neural Networks and especially LSTM– exploits additionally potential temporal and sequential dependencies at different data time-steps. We find that a sequential model of user behavior is a plausible choice for this binary classification, because such a model can capture the context of individual user behaviours, e.g. whether the user exploits opportunities to reveal pairs, even if the corresponding cards were revealed long ago. Thus, we train a sequential neural model based on LSTM [10]. For comparison, we trained also a static Linear Discriminant Analysis LDA model as baseline. To train those models, we need a large amount of training data episodes. In a future work, we aim at implementing a web-based version of our matching pairs game variants to enable collecting such a substantial amount of training data. In this paper, we use a cognitive user simulation based on Cognitive Memory Model CMM[11] to simulate substantial amount of HCI sessions with and without obstacles. In the next subsections we discuss the CMM simulator, the baseline static model LDA and the sequential model LSTM.

A. CMM-based Cognitive User Simulation

The Cognitive Memory Model [11] (CMM) is a computer program that utilizes the concepts of the ACT-R theory [12] to model the cognitive human memory. The CMM models

[1] The experiments have been done in two separate phases: memObs_noAdapt vs. memObs_LiMemAdapt and memObs_noAdapt vs. memObs_StMemAdapt. This explains why memObs_noAdapt does not show the same results in both comparisons.

memorized and forgotten items in human memory (revealed cards in our case) by weighting frequency and recency of their stimulations. The CMM has been successfully used to implement a generative model of playing matching pairs game by weighting the need for exploration of unknown cards and for exploitation of known pairs [13]. The CMM has a set of parameters (e.g. memory decay) which are optimized to model the human memory performance. In total, CMM has seven parameters which are randomly initialized and then repeatedly optimized using a genetic optimization algorithm. Those parameters can be further extended with application-based parameters, e.g. randomizing parameter in matching-pairs game to realistically simulate non-perfect human memory. Moreover, we introduced a similarity matrix to simulate similarities between cards in matching-pairs game. That is, given all the cards in a matrix as rows and columns, the diagonal will have only 1's where cards are identical to themselves, while other cells will contain values between 0.0 and 1.0 that define how similar to each other are the cards in the corresponding row and column. This similarity matrix is called then by the simulator for each revealed card to emulate human confusion happens especially when revealing those cards of red-greed shaded colors in the VISOBS_NOADAPT and VISOBS_VOICADAPT game variants. Since the cards show only colors (See Figures 1 and 2), we use the CIE1976 color model[2] to calculate the similarity values between those shown colors. A new parameter, `Similarity Decay`, is also introduced, which reduces the similarity effects while game is running: user may learn and adapt to such similar cards while the game is running, moreover, the similarity effects reduce continuously with less cards in the game after detecting pairs. All the parameters are optimized by a genetic optimization algorithm to best fit the reference real data, to which the simulated game sessions should be similar. This genetic optimization begins with initializing the parameters population randomly, and then the optimizer repeatedly updates that population by applying mutation and selection operations. For selection, we define two similarity metrics (Matching Score and Penalties) to compare between the simulated game sessions and their references (real game sessions). While `Matching Score` counts the number of matching pairs per round, `Penalties` assigns a cumulative punishment per round in case of revealing a card whose partner has been already seen before without picking up the pair. According to `Matching Score` and `Penalties` measurements, the best combination of memory parameters is selected as a memory configuration.

B. Baseline Static Model LDA

LDA is a linear classification model typically used to find a linear combination of given features to separate the corresponding data episodes into their classes. LDA, as a static model, only depends on such static features to classify data, without obtaining potential temporal dependencies between different time-stamps within a data episode (See next section V-C for more details about such temporal dependencies).
As mentioned in section V, the classes (labels) of our HCI application (Matching-pairs) are: `No Obstacle` and `HCI Obstacle`. Such a discrimination in LDA depends solely on user performance which can be defined via statistics calculated after finishing the game session. Such statistics can be also calculated at a specific time-stamp e.g. after 10 rounds (first revealed 20 cards) to examine the classification at that time-stamp. Similar to [4], we define the following discriminant features to discriminate game sessions based on user performance: "1) Number of cards left in the game. 2) Number of never revealed cards in the game. 3) Maximum number of times revealing the same card. 4) Number of rounds since game completion."

C. Sequential Model LSTM

For modeling sequential behavioural data in matching-pairs games, we employ an LSTM network. Our LSTM-based classifier consists of three layers: 1)LSTM input layer with 32 units (cells), 2)followed by a regularization layer (Dropout with rate=0.5) to avoid over-fitting 3) and finally a fully connected dense layer with soft-max activation acts as an output layer. Sequence items are passed consecutively to LSTM cells as consecutive time steps. The LSTM cells are interconnected where the output of one cell acts as an additional input to the next cell. Such a specific structure allows LSTM to store and retain information from previous time steps. To fit the model, we perform 500 epochs of Stochastic Gradient Descent with adaptive learning rate, $lr = \frac{0.1}{\#epoch}$.
The input consists of the ordered sequence of consecutively revealed cards. Each card is represented by two features: 1) The card's position in the revealing order of motives, 2) and the position of the card in the corresponding pair. For example: the first revealed card is always encoded with the feature vector (1,1), If the second revealed card shows the same motive, it is encoded with the vector (1,2) (second card of the first motive), otherwise with the vector (2,1) (first card of the second motive). The advantage of this feature representation is that it is invariant to the actual motives but still captures the temporal relationships within a sequence. In addition to this raw behavioural data, we calculate the four statistical features mentioned in Section V-B incrementally for each time step. In contrast to the LDA model, the LSTM model considers those features at each time step, while LDA only uses the manually defined features as one vector calculated after the end of the game.

VI. Evaluation and Discussion

We begin with the evaluation of HCI sessions with real users for different game variants: e.g. NOOBS_NOADAPT, MEMOBS_NOADAPT, VISOBS_NOADAPT, MEMOBS_LIMEMADAPT etc. After that, we discuss the similarity between HCI sessions with simulated and real users. Finally, we evaluate our classifiers regarding to their

[2] following the implementation `https://python-colormath.readthedocs.io/en/latest/delta_e.html`

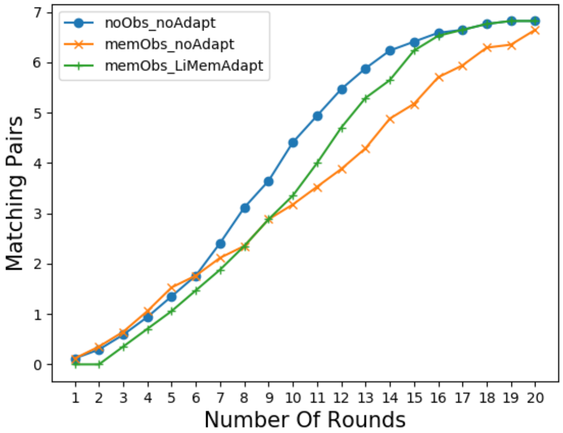

Fig. 3. Participants' mean performance analysis for: standard game, game with secondary task as memory obstacle supported/unsupported with a light UI adaptation

ability of detection of HCI obstacles and behaviour changes when applying light/strong UI adaptations.

A. Impact of Obstacles and Adaptations on User Behaviour

Figures 3, 4 and 5 show user performance under different conditions: game with/without obstacle and with/without UI adaptation. Statistical t-test shows, however, no significant difference ($p > 0.05$) between all game variants. This can be explained by the limited number of pairs used (7 pairs, 14 cards) and thus the short sequences of actions (selected cards) recorded. However, the aforementioned figures show different user behaviours for each game variant. Figure 3 shows three different user behaviours (captured as performance value per user action, i.e. Matching Score) for the noObs_noAdapt, memObs_noAdapt and memObs_LiMemAdapt game variants. While the secondary task (cumulative summation in memObs_noAdapt) deteriorates the HCI performance, the light UI adaptation (replay last round in memObs_LiMemAdapt) improves the HCI performance lightly, but the performance is still worse than the standard game noObs_noAdapt. Figure 4 shows similarly that the strong UI adaptation (replay all elapsed rounds in memObs_StMemAdapt) strongly improves the user performance. It becomes very similar to the performance in the standard game without obstacles noObs_noAdapt, despite that the secondary task (memory obstacle) still exists. While performance (matching pairs per round) is strongly improved with such a strong UI adaptation, it lasts long (Times needed in TableII). Moreover, Figure 8 shows that if we evaluate the user behaviour in memObs_noAdapt vs. memObs_StMemAdapt using Penalties measurement, we find the secondary task (cumulative summation) still impairs the normal behaviour of a standard game. In other words, Matching Score statistic can measure the performance well, however, it cannot detect whether an HCI obstacle exists or not, because it cannot distinguish between the normal user behaviour and the behaviour in presence of a memory obstacle and strong UI adaptation. To detect such obstacles and behaviour changes when applying UI adaptation, a behaviour-based classifier is required that can learn the behaviour rather than the performance of different sessions, see Section VI-C. For the vision obstacle, Figure 5 shows that while this visual obstacle makes it very difficult to distinguish the card (bad performance), an additional channel assistance (voice assistance: identifier spoken letters) tackles this obstacle. In contrast to the memory-obstacle case, the visual obstacle here is completely compensated by facilitating an additional non-impaired cognitive user ability.

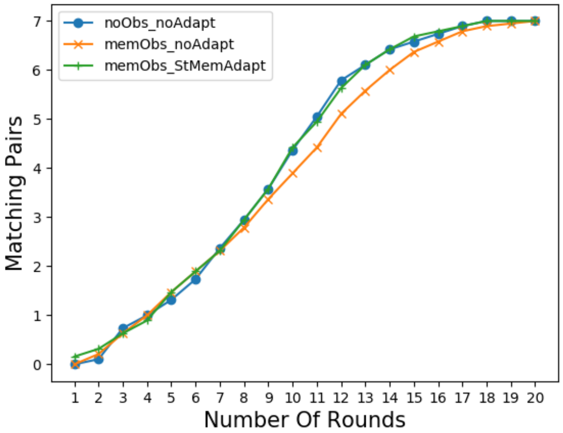

Fig. 4. Participants' mean performance analysis for: standard game, game with secondary task as memory obstacle supported/unsupported with a strong UI adaptation

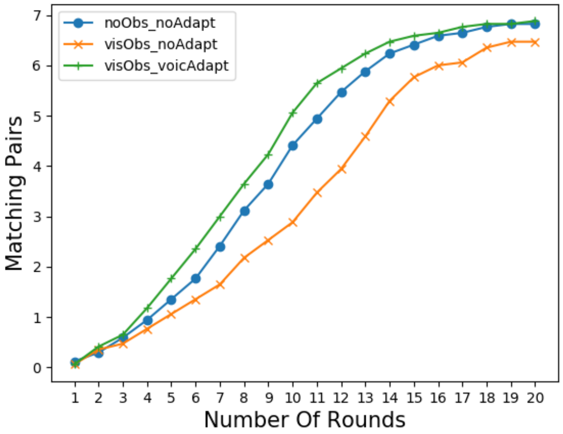

Fig. 5. Participants' mean performance analysis for: standard game, game with vision obstacle supported/unsupported with a voice UI adaptation

B. Plausibility of Cognitive User Simulation

As mentioned above, we need to simulate a substantial amount of plausible HCI sessions to train classifiers for the detection of HCI obstacles. In this section, we evaluate the similarity between our CMM-based simulated sessions

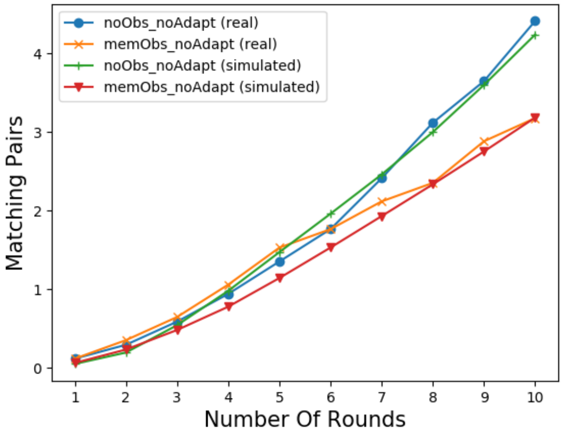

Fig. 6. Simulated sessions of standard game and memory-obstacle game follow the behaviour of the reference real sessions

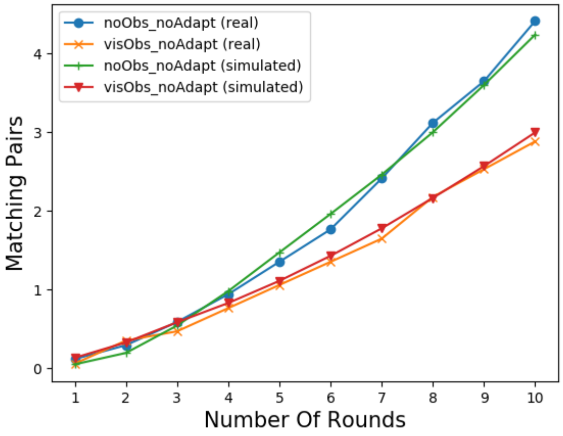

Fig. 7. Simulated sessions of standard game and visual-obstacle game follow the behaviour of the reference real sessions

and their reference real sessions. Figure 7 shows a close similarity for the whole first 10 rounds (first 20 cards revealed) between the simulated and real sessions in case of the NOOBS_NOADAPT and VISOBS_NOADAPT games. Figure 6 shows also a good similarity between the simulated and real sessions in case of NOOBS_NOADAPT and MEMOBS_NOADAPT games. However, we can see for the first rounds a difference between the lines of simulated and real memory obstacle sessions. This difference can be explained as the effects of the cumulative summation secondary task do not begin strongly in the first rounds in the real sessions. In contrast, the emulated color vision deficiency directly impairs the performance from the first round in real sessions.

C. Classifier Evaluation

In this section, we evaluate the baseline static LDA and the sequential LSTM classifiers from different perspectives. Both LDA and LSTM classify the HCI sessions as `no obstacle` or `obstacle`. We look at game prefixes after the 10th round (first 20 revealed cards). First, we train and evaluate the models according to cross-validation method using the 17 data sets of sessions and their corresponding simulated data (Train: 17000 simulated sessions per game variant, 1000 per participant log. Test: the 17 real data sets). For each fold, we also do 20 repetitions for the models for further stable results. In total, $17*20 = 340$ models have been trained, evaluated and stored for further evaluation with UI adaptations, see Table III.

TABLE III
ACCURACY OF HCI OBSTACLES DETECTION MODELS

Model/Classifier	Random Guess	LDA	LSTM
Memory-Obstacle Model	50%	**64.7%**	61.7%
Visual-Obstacle Model	50%	67.6%	**70.6%**

Results in Table III show that the static model LDA outperforms the sequential model LSTM for detecting a memory-obstacle, while the LSTM outperforms LDA for detecting visual obstacle. However, McNemar test shows that there is no significant difference between LSTM and LDA classifiers in both tasks (detection of memory-obstacle and visual-obstacles) with $p = 0.99 > 0.05$. Both classifiers beat the random baseline 50%. According to Müller-Putz et al. [15], we calculate the confidence interval of the random baseline using the Agresti equation [14] given $\alpha = 0.05$ as a significant threshold and $n = 34$ samples (17 participants, each with two sessions per task: `no Obstacle` and `Obstacle`). Consequently, we compare the accuracy of both LSTM and LDA with the resultant confidence interval of random baseline: $[0.33, 0.67]$. Thus, while we can detect different types of interaction obstacles, which influence a user's behavior in different ways (using either static classifier LDA or sequential one LSTM), this evaluation shows that only LSTM realistically outperforms the random baseline in the visual-obstacle detection task. We argue that the LSTM benefits from the good simulation performance for the visual obstacle, which generates behavior very similar to the reference real data, whereas the simulated memory-obstacle data has been shown to differ in the first few rounds compared to their reference real data (Figure 7 and Section VI-B).

To discuss how the LSTM does also exploit potential behavioural temporal dependencies in memory-obstacle sessions, we load the trained LSTM models and evaluate their ability of detecting changes of user behaviour when playing with different obstacles and different UI adaptations.

In contrast to the results of Table III, the Table IV shows that the visual-obstacle LDA and LSTM classifiers have similar accuracy, while LSTM outperforms LDA in memory-obstacle models. This is explained as we evaluate in the later Table how well LSTM and LDA detect changes in HCI with obstacles under different UI adaptations. We have seen that the UI adaptation used to tackle the visual obstacle was very suitable as it totally tackled the obstacle by facilitating another human cognitive ability (hearing).

TABLE IV
RESULTS OF OBSTACLES DETECTION AFTER UI ADAPTATIONS

Models	No-Obstacle	Obstacle
Static Baseline Classifier: LDA		
memObs_LiMemAdapt	29.4%	70.6%
memObs_StMemAdapt	67.8%	32.2%
visObs_voicAdapt	73.9%	26.1%
Sequential Classifier: LSTM		
memObs_LiMemAdapt	12.9%	87.1%
memObs_StMemAdapt	47.7%	52.3%
visObs_voicAdapt	73.2%	26.8%

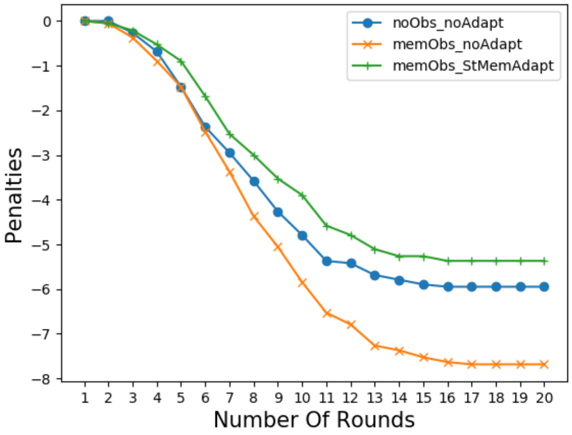

Fig. 8. Participants' penalties analysis for: standard game, game with secondary task as memory obstacle supported/unsupported with a strong UI adaptation

Thus, such sessions supported with voice assistance UI adaptation are classified only based on user performance, and this explains the similar accuracy between the baseline LDA as performance-dependent model and the sequential model LSTM. On the other side, the memory-obstacle UI adaptations (light and strong UI adaptations) do improve the performance but do not totally tackle the memory-obstacle because the cumulative summation as a secondary task still exists. In this case, the importance of exploiting temporal behavioural dependencies between different time steps in the tested data is highlighted. That is, although e.g. the performance of memObs_StMemAdapt games shown in Figure 4 looks like the same of noObs_noAdapt performance, Figure 8 shows in contrast different behaviours (captured as penalties, recall penalties measurement in Section V-A) between those different games. Consequently, the LSTM classified such sessions plausibly as memory-obstacle close to No Obstacle (52.3%) while the baseline static LDA –as a performance-dependent model– classified such sessions simply as No Obstacle (32.2% as memory-obstacle) because LDA only depends on performance which is clearly improved by such a strong UI adaptation, while LSTM exploits potential temporal dependencies between different time steps in a session for such a plausible classification.

VII. SUMMARY AND CONCLUSION

In this paper, we discussed memory-based and visual HCI obstacles and corresponding UI adaptations. We highlighted that a sequential model based on LSTM is a plausible choice for online detection of not only the HCI obstacles but also behaviour changes expected when applying UI adaptations. The LSTM model outperforms baseline models in the detection of visual obstacle with average accuracy 70.06%. Moreover, the LSTM plausibly matches the results of the subjective assessment regarding to the UI adaptations. The collected data, questionnaires and game variants description are available online in the Open Science Framework under the InteractionObstacles-MatchingPairs project: https://osf.io/bsudk/

REFERENCES

[1] Berka C, Levendowski DJ, Lumicao MN, Yau A, DvisObs_voicAdapt G, Zivkovic VT, Olmstead RE, Tremoulet PD, Craven PL. EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks. Aviat Space Environ Med 2007.
[2] Baldwin, Carryl L., and B. N. Penaranda. "Adaptive training using an artificial neural network and EEG metrics for within-and cross-task workload classification." NeuroImage 59.1 (2012): 48-56.
[3] Herff C, Fortmann O, Tse CY, Cheng X, Putze F, Heger D, Schultz T. Hybrid fNIRS-EEG based discrimination of 5 levels of memory load. In2015 7th International IEEE/EMBS Conference on Neural Engineering (NER) 2015 Apr 22 (pp. 5-8). IEEE.
[4] Putze, Felix, Mazen Salous, and Tanja Schultz. "Detecting memory-based interaction obstacles with a recurrent neural model of user behavior." 23rd International Conference on Intelligent User Interfaces. ACM, 2018.
[5] Bruno Sguerra, Pierre Jouvelot. "An Unscented Hound forWorking Memory" and the Cognitive Adaptation of User Interfaces. Communication pour la conference UMAP 19, Larnaca, Juin 2019.2019.
[6] P. A P Moran. 1958. Random processes in genetics.Mathematical Proceedings ofthe Cambridge Philosophical Society54, 1 (1958), 6071
[7] Jefferson, Luke, and Richard Harvey. "An interface to support color blind computer users." Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, 2007.
[8] Qaiser, Abu Zohran, and Muhammad Taha Khan. "Adaptive Interface for Accommodating color-Blind Users by Using Ishihara Test." arXiv preprint arXiv:1712.03329 (2017).
[9] National Eye Institute, https://nei.nih.gov/health/color_blindness/facts_about, 28 03 2019.
[10] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
[11] Proepper, Robert, Felix Putze, and Tanja Schultz. "Jam: Java-based associative memory." Proceedings of the Paralinguistic Information and its Integration in Spoken Dialogue Systems Workshop. Springer, New York, NY, 2011.
[12] Anderson JR, Bothell D, Byrne MD, Douglass S, Lebiere C, Qin Y. An integrated theory of the mind. Psychological review. 2004 Oct;111(4):1036.
[13] Putze F, Schultz T, Ehret S, Miller-Teynor H, Kruse A. Model-based Evaluation of Playing Strategies in a Memo Game for Elderly Users. In2015 IEEE International Conference on Systems, Man, and Cybernetics 2015 Oct 9 (pp. 929-934). IEEE.
[14] Agresti A, Caffo B. Simple and Effective Confidence Intervals for Proportions and Differences of Proportions Result from AddingTwo Successes and Two Failures. The American Statistician 54(4):280-288, 2000.
[15] Müller-Putz G, Scherer R, Brunner C, Leeb R, Pfurtscheller G. Better than random: a closer look on BCI results. International Journal of Bioelectromagnetism. 2008;10(ARTICLE):52-5.
[16] Putze F, Schultz T. Adaptive cognitive technical systems. Journal of neuroscience methods. 2014 Aug 30;234:108-15.
[17] da Costa Vieira, Pedro Cosme. "T-Test with Likert Scale Variables." (2016).