



# Improving the understanding of web user behaviors through machine learning analysis of eye-tracking data

Diana Castilla<sup>1</sup> · Omar Del Tejo Catalá<sup>2</sup> · Patricia Pons<sup>2</sup> · François Signal<sup>2</sup> · Beatriz Rey<sup>3</sup> · Carlos Suso-Ribera<sup>4</sup> · Juan-Carlos Perez-Cortes<sup>2</sup>

Received: 21 August 2021 / Accepted in revised form: 17 May 2023  
© The Author(s) 2023

## Abstract

Eye-tracking techniques are widely used to analyze user behavior. While eye-trackers collect valuable quantitative data, the results are often described in a qualitative manner due to the lack of a model that interprets the gaze trajectories generated by routine tasks, such as reading or comparing two products. The aim of this work is to propose a new quantitative way to analyze gaze trajectories (scanpaths) using machine learning. We conducted a within-subjects study ( $N = 30$ ) testing six different tasks that simulated specific user behaviors in web sites (attentional, comparing two images, reading in different contexts, and free surfing). We evaluated the scanpath results with three different classifiers (long short-term memory recurrent neural network—LSTM, random forest, and multilayer perceptron neural network—MLP) to discriminate between tasks. The results revealed that it is possible to classify and distinguish between the 6 different web behaviors proposed in this study based on the user's scanpath. The classifier that achieved the best results was the LSTM, with a 95.7% accuracy. To the best of our knowledge, this is the first study to provide insight about MLP and LSTM classifiers to discriminate between tasks. In the discussion, we propose practical implications of the study results.

**Keywords** Machine learning · User experience · Usability · Eye-tracking · Long short-term memory recurrent neural network · LSTM · Multilayer perceptron neural network · MLP · Gaze · Trajectories

---

✉ Diana Castilla  
Diana.Castilla@uv.es

<sup>1</sup> Departamento de Personalidad, Evaluación y Tratamientos Psicológicos, Universidad de Valencia, Valencia, Spain

<sup>2</sup> Instituto Tecnológico de Informática (ITI), Universitat Politècnica de València, Valencia, Spain

<sup>3</sup> Departamento de Ingeniería Gráfica, Universitat Politècnica de València, Valencia, Spain

<sup>4</sup> Departamento de Psicología Básica, Clínica y Psicobiología, Universitat Jaume I, Castellón de La Plana, Spain

# 1 Introduction

## 1.1 Eye-tracking in human–computer interaction

With the exponential growth of information and communication technologies (ICTs), the understanding of the user's experience (UX) has become increasingly important in research and applied settings. A frequent concern in research into human factors and engineering has been the ability to capture human cognitive abilities and behavior when interacting with systems using a scientific methodology. UX, however, is a complex construct composed of both rational and emotional elements of judgment, which makes its quantification challenging (Sauro and Lewis 2016). Traditional measurements of UX, such as task complete ratio and user opinion, to name some examples, can be informative of problems with a technology (e.g., whether the user completed a task or felt the system was hard to use). However, these assessments fail to indicate why such challenges occur and how to fix them (Ghasemifard et al. 2015). Qualitative evaluations of the end user's opinions might be highly informative, but they can be biased due to social desirability or task-related stress factors, among others. The search for methodologies that complement this qualitative data and allow a more accurate understanding of the user experience remains an unsolved challenge.

Eye-tracking technology might be a feasible complementary tool for the assessment of UX. Infrared eye-trackers allow to capture the interaction between visual stimuli and the user using an infrared light that is reflected by the cornea and the center of the pupil. Infrared eye-trackers are the most common type of eye-trackers, although other technologies not based on infrared light have also been proposed (Singh and Singh 2012). Irrespective of the type of eye-tracker used, these tools might be particularly useful in the field of UX because they allow recording not only the pupil's size, but also the user's gaze at the interface. Particularly, eye-tracking data provide instantaneous measurements of the positions and durations where the eyes are looking at during specific tasks or trials. Importantly, the analysis and conclusions that can be obtained from eye-tracking data cannot be extracted from other behavioral or physiological measures. In terms of UX, this information can be useful to understand the cognitive processes that are occurring during the execution of the experimental task (Beesley et al. 2019), such as the elements that are looked at first or during longer periods of time. Ultimately, this can be used to detect user–technology interaction problems (e.g., users that are interacting visually with an element in an unexpected or in an inadequate manner), which might help find appropriate solutions, such as changes in the technology or training in the users (Rosch and Vogel Walcutt 2013).

Eye-tracking devices have been used as complementary evaluation tools in usability and user experience studies since the 1970s (Mele and Federici 2012). However, their widespread utilization has been traditionally difficult due to challenges in data analysis and some limitations of the technology itself. For example, the first eye-trackers required restricting the user's head movement, which is problematic in terms of user acceptability. In the last 10 years, the development of novel analytical procedures

and the emergence of modern, user-friendly, and economic eye-trackers has made eye-tracking a really attractive tool for researchers interested in UX (Schall and Bergstrom 2014).

## 1.2 Quantitative analysis in eye-tracking and its current limitations

While acknowledging the potential of eye-tracking technology for the assessment of UX, recent research has pointed to a number of challenges to make the most out of the data it offers (Zaki and Islam 2021).

Two important eye-tracking parameters that have been related to UX are the fixations and the saccades. A fixation can be defined as a relatively stable gaze at a target. The exact time period needed by the human visual system to perceive an element varies depending on the kind of object. Consequently, the temporal threshold required to identify a fixation also changes depending on the target (Djamasbi 2014). Many web studies use a 100 ms threshold (i.e., Buscher et al. 2010), although other web studies have proposed a longer period of 300 ms as a minimum time to identify features that need a longer sustained attention (Djamasbi et al. 2010). Saccades refer to rapid eye movements that happen between fixations, that is, when the user shifts attention from one target to another (Asan and Yang 2015). Based on these two parameters, additional valuable parameters have been established. For example, scanpaths are derived from fixations and saccades and represent the path or ordered sequence of fixations and saccades made by a user's eyes on an AOI (Area of Interest) space in the graphical interface where a relevant stimulus is located (Roa-Martinez and Vidotti 2020). Scanpaths can help researchers to identify patterns in the eye-tracking data and provide a deep understanding of how people use and interact with their environment. This has been especially relevant in the fields of psychology, as well as in human–computer interaction studies. For example, scanpaths allow researchers to detect regions of interest in websites (Eraslan et al. 2016b), mental processes behind the observation of an image (Król and Król 2020), or participants who might behave differently due to a specific physical or mental condition (Eraslan et al. 2019). In addition, people with different cognitive styles differ in the way they visually search for information (Crosby and Peterson 1991), so the participant's cognitive style has been inferred from eye-tracking data (Raptis et al. 2017).

In most cases, basic quantitative analyses based on techniques provided by the eye-tracker manufacturer (e.g., heatmaps—graphical representation of the areas where the eyes are fixated most often during the task, series of fixations, means and standard deviations, and AOIs) can be applied to extract relevant temporal information from the fixations (Drusch et al. 2014). This temporal analysis of fixations represents valuable information, such as the points and moments where the eye movements pause. In addition, the analysis of saccades and heatmaps provides a useful graphical representation of the areas where the eyes are fixed most often during the task, which allows to define or infer areas of interest based on user behavior.

In addition to the aforementioned information integrated into the commercial tools for eye-tracking, a number of open-source and free applications are also available to analyze eye gaze data, including GazeVisual (Kar and Corcoran 2019), ScanGraph

(Dolezalova and Popelka 2016), OGAMA (Voßkühler et al. 2008), GazeAlyze (Berger et al. 2012), or EyeMap (Tang et al. 2012), to name some examples. Therefore, the current analytical armamentarium for eye-tracking systems is diverse.

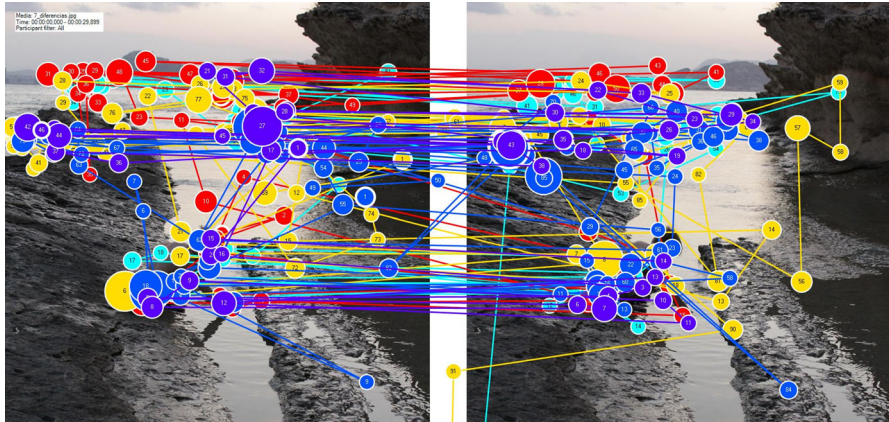
Complementary analyses to those provided by these automatic tools also exist and include the computation of different scores. For example, in a study in which participants had to complete a visual paired comparison recognition memory task, a novelty preference index was defined as the percentage of time that the participant spent trying to find the new image in comparison with the familiar one (Bott et al. 2018). Other studies have proposed additional parameters extracted from the eye movement trajectory (Kao et al. 2018), including the count of fixations and saccades, the percentage of fixation duration for specific regions, and the alternation of visualization between specific pairs of regions. In the study by Kao et al. (2018), these features were used as the input to a support vector machine to perform a linear classification to identify the ability of the readers during a reading behavior experiment. Altogether, there is little homogeneity when it comes to additional analyses that are recommendable when using eye-trackers. Also importantly, the cutoff scores for such parameters are yet to be determined and their values appear to be highly dependent on the experimental design used.

Some attempts to establish robust statistical approaches to eye-tracking, particularly in fixation data, do exist. One example of this is the lag sequential analysis (Tsai et al. 2018), which has been applied to explore visual-attention transitional patterns (between previously defined areas of interest) during learning processes and to compare the observed probability of a behavioral event and the expected one. Additional techniques have been proposed and applied to analyze and compare temporal series of areas of interest (Eraslan et al. 2016a). Some of these techniques have been applied to estimate the similarity/dissimilarity between pairs of webpage user scanpaths, to detect patterns and transition probabilities between different visual elements, or to identify a common scanpath for multiple scanpaths.

Not all studies, however, rely on fixation series for their analysis of UX with the eye-tracker. For example, complementary information from the eye-tracking signals, such as pupil dilation and spontaneous eyeblink rate, has also been used in the literature. Pupil dilation has been utilized as an indicator of subjective task difficulty, mental effort, and neural gain, while eyeblink rate has been associated with learning and goal-directed behavior processes (Eckstein et al. 2017). Open-source software has been developed to analyze pupil dilation combined with gaze position, which facilitates the dissemination of this procedure (Geller et al. 2020).

As highlighted in the previous paragraphs, there is a significant gap between the features offered by the eye-tracking commercial software and the variety and relevance of user behavior patterns that could be analyzed with eye-tracking data. In fact, while great sophistication has been achieved by eye-tracking devices and data analysis software, researchers (even those with experience in this kind of metrics) still visually inspect eye-tracking data to extract conclusions about additional behavioral patterns, which means that a significant amount of data is based on qualitative analysis (Papavlasopoulou et al. 2020).

This inspection is done manually to describe valuable information about the user's execution in a number of areas. For example, visual analysis is frequently implemented



**Fig. 1** Example of multiple scanpaths displayed at the same time

to describe an erratic pattern in the saccades during a reading task or patterns in the execution of an attentional task (Ioannou et al. 2020). A visual inspection of the data is problematic because it is very time-consuming for the researcher, who will have to visually review every second of the fixations and saccades made by all the users and will have to interpret whether there is a comparable pattern among users. This work has to be done user by user because when there are multiple scanpaths from different users, it is difficult to distinguish between them. This can be observed in Fig. 1, where the scanpaths of different users during the performance of a task are represented.

In addition to this high investment in time, the visual and manual inspection of this complex data also represents a problem in terms of human error during data inspection and interpretation bias. In this sense, Norgaard and Hornbaek (2006) found that the expectations of UX professionals influence the UX issues analyzed. Therefore, a manual inspection requires the task to be repeated independently by another researcher, thus boosting the required investment of time and resources.

### 1.3 Machine learning approaches for eye-tracking tasks

Prior to the broad availability of eye-tracking devices that incorporate and simplify eye detection and data extraction, machine learning techniques were used to detect eye data from raw video images of faces/eyes. Several works have used machine learning to explore eye-related data, including the detection of eyes' pupils (Colombo et al. 2007; Duchowsky 2003), saccades and fixations (Urruty et al. 2007), smooth pursuit movements (Vidal et al. 2012), and scanpaths of users (Eraslan et al. 2016b; Haass et al. 2016).

Machine learning techniques have emerged as potential solutions to analyze the aforementioned scanpath patterns that are traditionally investigated with visual inspection. The identification of patterns in scanpath data has been particularly popular in the detection of tasks based on eye movements. For example, Yarbus's pioneer study

on eye-tracking postulated that the task given to a participant has a very large influence on the participant's eye movement (Yarbus, 1962). This section will review how machine learning techniques have been used over the past decade for different task classification, in order to analyze potential techniques that could be promising based on the kind of data and tasks available.

The use of machine learning has already resulted in improved metrics in research into scanpaths and has been applied to study whether the task performed by a participant could be predicted based on the participant's eye movements. For this kind of analyses, Greene et al. (2012) made one of the first attempts to classify a series of tasks based on the participant's eye movements. This resulted in a poor accuracy of 25.9%, which was not significantly over the chance level. The authors used a linear discriminant classifier and statistic features from scanpaths, which prevented them from using spatial and temporal data. This study was replicated with different classifiers, such as kNN and boosting, and including spatial information from the first five fixations (Borji and Itti 2014), which resulted in a slight improvement of the classification accuracy (34.12%). Yet another work analyzed Greene's dataset under a different classifier, namely a Fisher Kernel, to keep the temporal information within the time series (Kanan et al. 2014). The average accuracy for task prediction, however, was again low (33.1%). In the work by Haji-Abolhassani and Clark (2014), the authors highlight the lack of explanatory power of the summary statistics used by Greene et al., which have been shown to be unreliable in task inference (Castelhano and Henderson 2008). It was not until the work by Haji-Abolhassani and Clark (2014), when the authors replicated Greene's study using hidden Markov models as classifiers and obtained an average accuracy of 55–65%, which was significantly above the chance rate for the first time.

In addition to replicating the study by Greene et al. (2012), other examples exist using machine learning for the classification of eye-tracking data. For example, Coco and Keller (2014) evaluated data from 74 participants using artificially generated images with three different types of regression models and obtained an 88% classification success rate with one of them, that is, the support vector machine classifier. Other researchers have also compared different classifiers using machine learning, such as linear discriminant, quadratic discriminant, linear Gaussian naïve Bayes, and nonlinear Gaussian naïve Bayes (Kardan et al. 2015, 2016). More recently, an SVM approach was compared against an AdaBoosting method with decisions trees in another classification task, with the latter one obtaining better results (Kumar et al. 2019). In a similar fashion, Król and Król (2020) evaluated eye-tracking data from 148 participants performing 4 different tasks in two versions each: free viewing versus fixation on specific point. They also compared two different dimensions: semantic processing (shallow vs. deep) and spatial processing (global vs. local). For each trial, the computed features were the gaze median point, dispersion, and screen coverage of three different periods, using a feed-forward neural network classifier for task prediction. Another interesting approach to analyze eye-tracking data with ML techniques can be found in the study by Pfeiffer et al. (2020). This study investigated the user behavior in virtual reality (VR) stores by comparing the results of the user behavior in physical stores using an interesting parameter, that is, the variance of fixation duration as a predictor in life-size environments. Using ensemble methods, they obtained an average prediction accuracy

during the search process of 90% in VR and of 92% in physical contexts, which is a remarkable result. Their results demonstrated the utility of eye-tracking data not only to adapt the information from online systems, but also to use it for sale purposes with mobile eye-tracker devices.

Other previously discussed quantitative approaches have focused on specific statistical analyses to compare between temporal series of fixations in regions of interest, such as detecting similarities between temporal series or evaluating probabilities of transition between areas of interest (Tsai et al. 2018; Eraslan et al. 2016a). This kind of information is not always relevant for the researcher, who usually relies on a visual inspection to identify patterns in recorded scanpaths during the performance of specific tasks. However, machine learning approaches have been found suitable to analyze scanpath temporal series and to classify them between different states depending on the task that is being performed by the participant. Accuracy levels range between 29.5% in the first studies in this field (Greene et al. 2012) and 80.3–95.4% in more recent investigations (Kumar et al. 2019). Further works are needed to move forward in the use of these technologies for scanpath analyses. Specifically, research should focus on the improvement of accuracy, on determining the best way to prepare the input data, and on identifying the most appropriate machine learning tools for each practical application.

As can be seen, neural networks have not been widely used in scanpath analysis. Moreover, whether spatial or temporal, the extraction of relevant features contained in scanpaths remains an open problem. This is why this paper compares three different classifiers, namely random forest (Lepetit and Fua 2006), multilayer perceptron or MLP (Rosenblatt 1958), and long short-term memory neural network or LSTM (Hochreiter and Schmidhuber 1997) for the analysis of 6 different and common web tasks based on users' eye-tracking movements. On the one hand, the first two classifiers use spatial information, while the last also includes temporal information. On the other hand, the first classifier is not a neural network but a set of decision trees, while the last two are neural networks. To the best of our knowledge, the proposed MLP and LSTM classifiers have not been previously evaluated in the literature for this purpose. The main objective is to determine whether these classifiers would allow to distinguish between behaviors such as reading, searching for web elements, fixing one's attention to an element, or comparing objects in a web interface and to propose its implementation as a simple parameter in commercial software.

## 1.4 Building a new quantitative model

Task success is the most common usability metric that it is calculated for almost any usability study that involves tasks. In this context, performance is defined as the degree in which the users can complete a task or a set of tasks successfully (Tullis and Bill 2013). The tasks are designed around the system goals and frequently represent the regular interactions with an application. In this sense, reading or comparing items is a frequent task in different contexts. For example, comparing items is a key action in online shopping or neurorehabilitation systems and reading a large amount of text is a key action in online newspapers, e-learning platforms, or online psychological

treatments. The scanpaths captured by eye-trackers provide a valuable information about the user's performance in these kind of tasks, because in a reading task, the performance of the user is measured in terms of how much the user's performance matches the expected trajectory of a reading (with a downward zigzag shape).

However, the analysis of scanpath patterns is a time-consuming procedure that is currently performed by hand. The usability expert must inspect visually all the pages of each user to determine which ones deviate from the standard behavior. For example, if 30 users completed a 20 min reading task, the UX professional would invest a large amount of time to inspect the user behavior (e.g., stopped reading in the middle of the text, or got distracted by another element and came back later, read diagonally without paying real attention, i.e., without reading the sentences until the end, etc.), plus the time to collect evidences about the findings, such as screenshots or notes. Furthermore, to avoid the risk of bias, this analysis should be repeated by another professional. To be more concrete with the example previously given, the assessment task must be carried out manually and currently a study involving 30 users with an average of 20 min per session could require more than 25 h of work. As described during the introduction of this text, eye-tracking data can be analyzed using machine learning techniques. To do so, however, specific knowledge of this discipline is required. Unfortunately, this knowledge is not available to all researchers in the UX field, which includes professionals from very different areas, such as engineering but also psychology, library science, and graphic design.

This work describes the implementation of different machine learning classifiers that could allow UX researchers to quantitatively estimate the degree to which a web task shows a pattern in its scanpaths. In doing so, we will also propose a way to integrate in future works this type of analysis into eye-tracking software, so that professionals without machine learning knowledge could benefit from this quantitative model.

Some web behaviors involve a repetition in the scanpath presenting a clear pattern. The most common tasks of web users include attentional and reading skills (for example, information search or shopping). An attentional task shows a pattern of large fixation in an object or a pattern of alternative fixations in two objects that are compared to each other, like a zigzag pattern. A reading task has a pattern of horizontal scanpaths connected to each other in a clear zigzag pattern. On the other hand, a website reading task might even be somewhat similar to free web browsing or reading, because it is possible to find patterns of reading but also patterns of free web browsing. Our main interest is to distinguish product comparison patterns and some forms of web reading behaviors from free web browsing. This is important because if we could automatically distinguish and quantify these behavioral patterns within a web page, we would be able to make significant progress in the analysis and design of user experience.

For example, when buying a product, the consumer reads its description and compares several images or products in the purchase-decision process. The scanpaths emerged in this process could tell us how to improve the webpages to optimize information depending on the type of consumption. Other areas, such as online learning or telepsychology, where the users have to read a large amount of information, could benefit from this innovation. To give a few examples of its applicability in future works,



knowing whether the information is read entirely or not or revealing the reading pattern followed by the user when other objects are near (like images) could be crucial to optimize the design of these systems. Further application possibilities could be related to systems that react in real time to the user's behavior. In this sense, the researcher could set the optimal task performance and the model could compare, in real time, whether the user is close to that result, so that the system would make decisions based on that (Gual-Montolio et al. 2022). For example, the system could offer cognitive rehabilitation or learning exercises based on task performance, thus allowing the user to progress in real time.

With this purpose in mind, we designed 6 tasks to imitate different behaviors of web users, such as attentional tasks, reading text, and free surfing in a web page.

To build the required quantitative model, this paper proposes the evaluation of a machine learning approach to build a discriminant model that will be designed to classify between six different tasks based on the user's eye-tracking movements. This machine learning classifier will use the information provided by an eye-tracking device to train different classifiers, thus evaluating their accuracy for the proposed tasks. To maximize the precision of our classification system, the evaluated classifiers will use temporal and spatial information from the user's eye-tracking scanpaths. Taking the type of information provided by the eye-tracker and previous works in the area, we will validate our hypothesis against three important classifiers, namely the random forest, MLP, and LSTM. We will apply a Trace-Segmentation for the normalization and comparison of scanpath sequences.

## 2 Materials and methods

### 2.1 Design

The study consists of a within-subjects experiment where 30 users have to perform six different tasks that simulate specific behaviors in web sites. Their performance will be assessed using the task analysis method (Hackos and Redish 1998), and the resulting scanpaths will be analyzed with three machine learning techniques, namely random forest, MLP, and LSTM. The tasks have been chosen based on six common web tasks, that is attention to an object, attention distinguishing between two objects, reading (only text, text with an image, and free reading on a website), and free surfing on a website. The tasks were ordered and assigned randomly to each participant to avoid a carry-over effect of task fatigue or learning.

### 2.2 Scanpath classification

Given a scanpath  $\vec{x}$  and being  $C$  the set of classes, the goal is to predict the class of the scanpath  $c_p$ , given in Eq. 1:

$$c_p = \underset{c}{\operatorname{argmax}} p(c \vee \vec{x}) \quad (1)$$



**Fig. 2** Example of a successful outcome in attentional task 1

To achieve this, a model must be trained to learn the distribution  $p(c \vee \vec{x})$ . This is done by training the classifier  $f$  and minimizing the loss  $L(X, Y)$  over the training data  $X$  and their respective true classes  $Y$ , as indicated in Eqs. 2 and 3:

$$\operatorname{argmin}_{\theta} L(X, Y) \quad (2)$$

$$L(X, Y) = -\sum_{x,y} \sum_{c=1}^y y_c \log f(x)_c \quad (3)$$

This work also assumes that every task is distinguishable from the other using the features provided by the scanpath. Therefore, we expect that this model will be efficient to detect routine web tasks such as reading, attention to an object, or attention comparing two objects.

## 2.3 Tasks

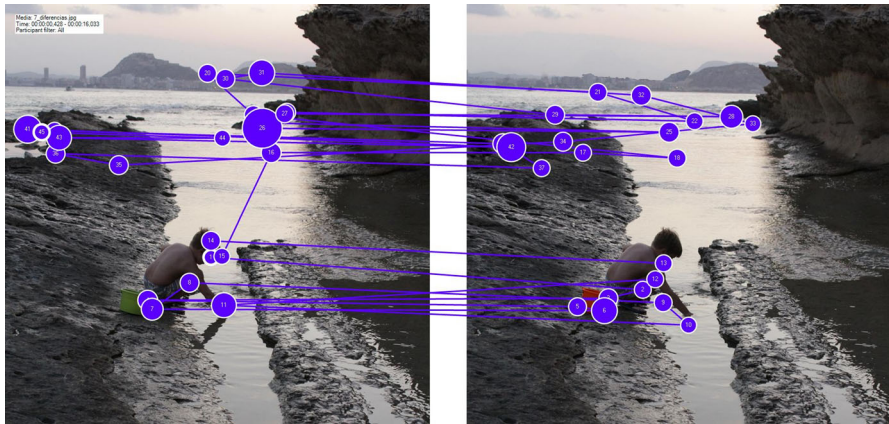
Six simple tasks were designed to represent different reading or attentional behaviors of web users, as follows:

### 2.3.1 Task 1: attentional task 1 (circle)

The attentional task 1 displays a single stimulus in the center of the screen (a circle). The participant is asked to stare at it for 6 s. Given the nature of the task, all fixations are concentrated in the center of the screen if the task is performed successfully (see Fig. 2). This task was designed as a checkpoint to check the user's ability to perform a simple attentional task.

### 2.3.2 Task 2: attentional task 2 (differences)

This task forces participants to shift attention by comparing two images that are almost identical except for 7 differences. The participants are asked to find at least 3 of these



**Fig. 3** Example of a successful outcome in attentional task 2

differences. This task was designed to represent the behavior of a user when comparing the characteristics of a product or item. In this task, the fixations alternate between one image and the other showing a random zigzag pattern saccade, as reflected in Fig. 3.

### 2.3.3 Task 3: reading task 1 (text)

The reading task 1 shows a paragraph. In the reading task, the saccade also draws a scanpath with a zigzag pattern, but ordered in descending order from the first line to the last, as shown in Fig. 4. The participants are asked to read the text once only. It is important to mention that a top-down comparison of two products can be very similar to a reading task.

### 2.3.4 Task 4: reading task 2 (text + image)

The reading task 2 shows a paragraph plus an image next to it. The participants are asked to read the text once. No instructions are given regarding the image. This is done to explore whether the visual stimulus next to the text (the image) serves as a distractor and mediates a change in the reading pattern. In this reading task, a scanpath with a zigzag pattern sorted in descending order from the first line is also expected (see Fig. 5).

### 2.3.5 Task 5: reading task 3 (searching in a web page)

The reading task 3 shows an educational webpage. The participants are asked to read the text during 1 min. The web context opens the possibility to read only small parts of the text without an established order. For this reason, in the reading task 3 a scanpath pattern with small zigzag reading blocks without a predefined order is expected (see Fig. 6). Sector changes are also expected without a predefined logic. This task represents a combination of the reading patterns described in tasks 3 (Fig. 4) and 4 (Fig. 5) and the random scanpaths in task 6 (Fig. 7).

Es una familia, salga bien o mal, me ha ocurrido ahora por casualidad.  
 Cerca de unos prados que hay en mi lugar, pasaba un borrico por casualidad.  
 Una mañana en un campo que me quedaba un poco olvidada por casualidad.  
 Acordé con el dueño del campo que me lo cediera por casualidad.  
 En la mañana se me olvidó de ir, y se me olvidó la posibilidad.  
 "¡Oh! ¡Dijo el borrico, "¡Que se me olvidó ir y dijo que se me olvidó ir!"  
 Sin reglas de juego, borricos hay que se me olvidan por casualidad.

Media: Esta tabquilla\_Page\_1.png  
 Time: 00:00:00.735 - 00:00:23.443  
 Participant: Wier: J6

Fig. 4 Example of a successful outcome in reading task 1

Media: Por entre unos mataos\_Page\_1.png  
 Time: 00:00:00.630 - 00:00:26.679  
 Participant: Wier: J6

Por lo que me metí a seguir de perros, -no diré cosas, volaba un conejo.  
 Dije mal de la casa y me compere, y le dije: "Detente, ¿qué me dices?"  
 "¿Qué me dices de ser por los perros, sin salir a jugar. Dos perros galgos me vienen siguiendo."  
 "Sí -replica- por los perros y los perros. Pero yo me digo: "¿Pues qué es eso?" "Por los perros."  
 "¿Qué? ¿Podemos decir? "Sí, como me dices."  
 "Galgos y muy seguros, tienen un ojo."  
 "Son perros, voy a que me dices de los perros."  
 "Son galgos, te digo. "Digo que por los perros."  
 En ese momento, llegamos a los perros, me desquicé con mis tres conejos.  
 Los que por culpa de los perros me dejaron lo que importa, llévase eso." nlo.

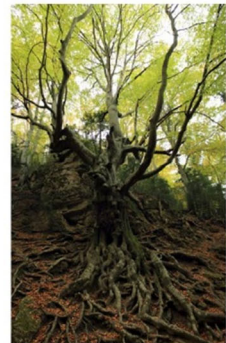
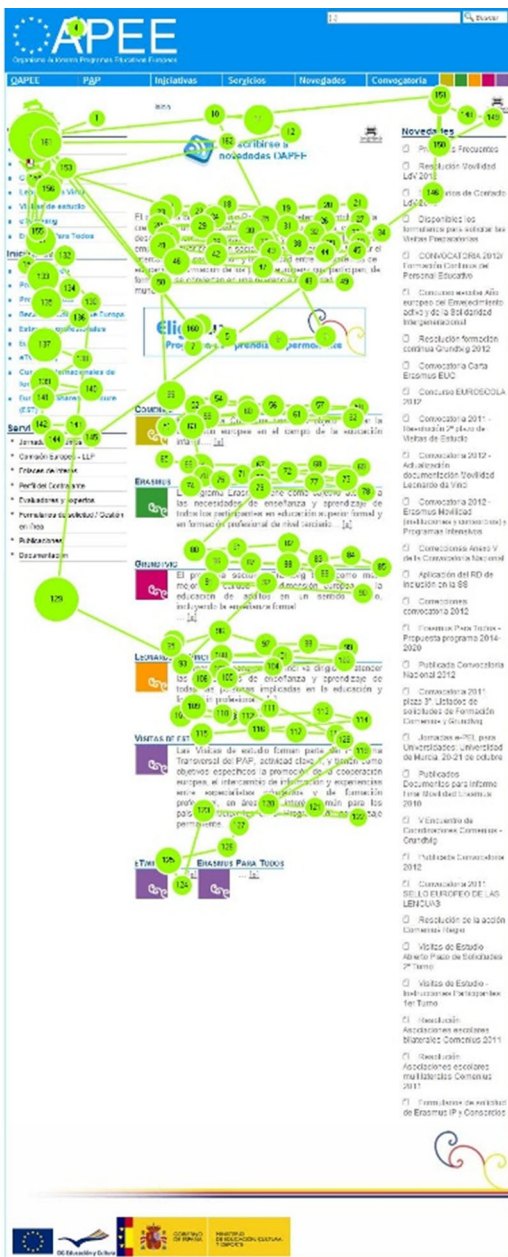


Fig. 5 Example of a successful outcome in reading task 2

Media: <http://www.cerme.es/temas/tema10.html> (PDF)  
 Title: 00000000-0000-0000-0000-000000000000  
 Publication Year: 0



Todos los derechos reservados. [www.cerme.es](http://www.cerme.es)  
 (00000000) (000000) (000000)

Fig. 6 Example of a successful outcome in reading task 3

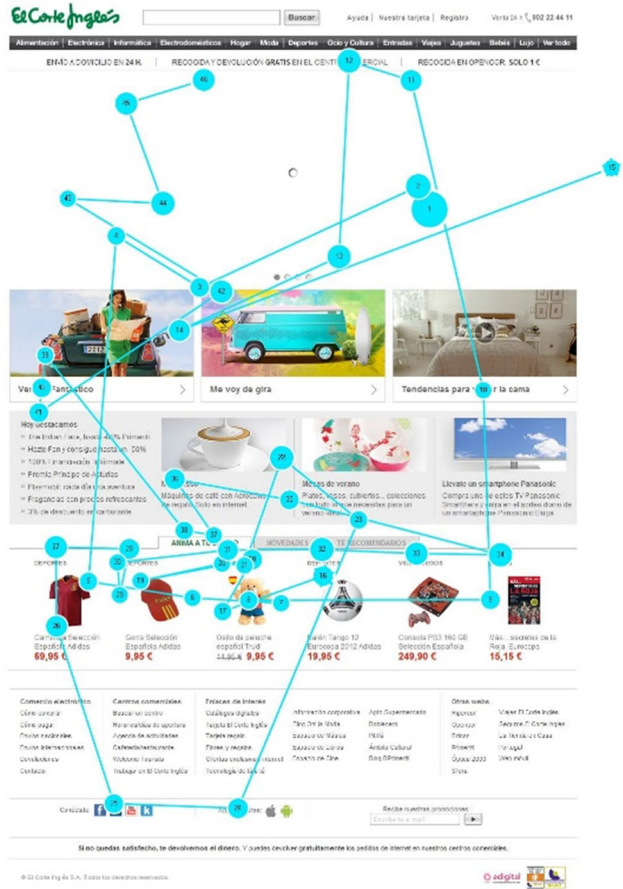


Fig. 7 Example of a successful free surfing result

### 2.3.6 Task 6: free surfing in a web page to shop for a specific product

Task 6 shows a shopping webpage. The participants are asked to search freely for a product of their choice. Given the web context of this task, a random scanpath without any pattern is expected (see Fig. 7). This task was designed to discriminate an unstructured reading on the web from an exploratory behavior without reading.

Given the within-subjects design of the study, the tasks were presented in different orders across the participants to avoid a carry-over effect. There were six order possibilities:

- (1)123456 (2)234561 (3)345612 (4)456123 (5)561234 (6)612345

## 2.4 Participants and procedure

Participants were 30 undergraduate students aged between 21 and 45 years (mean age = 29.93 years, SD = 6.07; 43.3% men and 55.7% women). All the participants were volunteers recruited at a large university in Spain. No compensation was offered for participation. The inclusion criteria were being 18 years of age or older, having the audio, visual, and motor abilities to interact with the system, and not having undergone a cataract surgery (because the intraocular lens placed in this operation interfere with data collection with the eye-tracker).

The study was conducted in the Laboratory of Psychology and Technology (LabP-siTec) of the Jaume I University (Spain). After signing the informed consent form, the participants were randomly assigned to one of the six possible orders and were informed about the general rules of the experiment, which included the need to remain seated and not moving the chair until the end of the task. The instructions for each task were displayed in the middle of the screen by the Tobii Studio Software. At the beginning of the experiment, the gaze of each participant was calibrated and validated with the Tobii Studio Software, which presents a grid of 9 stimuli points at gaze angles  $\leq 18^\circ$ . Next, the participants received the specific instructions for each task in the order assigned during the randomization.

## 2.5 Dataset

For this study, we used an eye-tracker with a 23" screen (1920 × 1080 pixel resolution and 16:9 aspect ratio) and an integrated camera. This device has an unobtrusive eye-tracker that allows head movement with a sample rate of 300 Hz (Tobii TX 300, n.d.). This means that eye movements (saccades and fixations) can be recorded without using a chinrest. The user has to maintain a distance from the screen of 20–31" (50–80 cm) and has a freedom of head movement of 15 × 7" (37 × 17 cm). This freedom, however, implies that the recorded signal might be lost during movements, for example, if the user looks away from the screen. To minimize this problem, the validity of the signal was controlled in this study. The software used for data recording was the Tobii Studio Enterprise edition. Both this tool and the usual eye-tracking measurements display a precision record of signal validity. When the recorded signal is lost (because the user blinks or looks at the keyboard instead of the screen), the software records a validity of 4, which is interpreted as missing data. The webcam also records the user's behavior, which allows to check why the gaze record has been lost during missing data periods (validity = 4).

Eye-tracking data typically consist of a large number of fixations in the screen coordinates ordered by a timestamp of when the gaze data were collected. The data were collected in a raw format. The data collected for this experiment included the position (vertical and horizontal) of the gaze point for each eye separately (left and right), as well as the timestamp. Because this eye-tracker model allows free head movement, data loss is naturally expected in situations where the user loses eye contact with the screen due to head movement (e.g., sneezing or looking at the keyboard while writing). Data loss is also expected when participants blink. For this reason, the validity of the gaze data for both eyes was also gathered (scored from 0 to 4). The validity is 0

when the tracking quality is good and 4 when data cannot be found by the eye-tracker (as already explained, because the user looks away from the screen or blinks).

The full dataset contains a total of 180 trials, 30 for each task, which are divided across the six tasks uniformly. Approximately 7.47% of the eye-tracking data (141,289 samples out of 1,891,285) was lost due to the free movement of the head, and the natural behavior of the user during the task (such as looking at the keyboard while typing or blinking).

For each scanpath, the eye-tracker system provides, with a rate of 300 Hz, the following features: coordinates  $X$  and  $Y$  of each eye, the quality of each eye's coordinate detection, the eye-tracker's timestamp, the timestamp for the recording, and the user ID.

## 2.6 Data quality and pre-processing

Given a good frame rate and based on the already described validity index, low-quality detections could be consequently filtered out from the raw data as the number of remaining points was significant enough.

Raw data also contain the coordinates of the saccade for each eye. There were many options to combine the information, but only the left eye coordinates were selected. Although most eye-trackers register binocular data, researchers may analyze the coordinates of one eye only, the average of both coordinates, or both coordinates separately depending on the purposes of the experiment and the acceptable error (Hooge et al. 2019). We experimentally checked that the difference between the saccade coordinates for both eyes could be reduced to a single eye without losing precision.

### 2.6.1 Delta transformation

Absolute coordinates may not be as useful for the algorithm as the relative movement of the saccade. Hence, we transformed the initial sequence of absolute coordinates of the image to relative movements between two consecutive points of the trace. This reduces the length of the sequence by 1, so we modified the value of the  $k$  in trace segmentation accordingly to keep the same length for all the experiments.

### 2.6.2 Normalization

Normalization of the data is required for some machine learning algorithms such as neural networks. These algorithms may use nonlinear activation functions which heavily rely on low range input values, as the larger the input values to the function, the lower their derivatives would be for training. Normalization is performed using the size of the largest image of the 6 tasks. This value is used as a reference to project all points into the range  $[0-1]$ , which is acceptable for the algorithms. For instance, if the task with the largest image has a resolution of  $1000 \times 1500$  pixels, widths are divided by 1000 and heights by 1500 in all the traces.



## 2.7 Machine learning algorithms

Each scanpath must be represented by a set of features (numerical values) of a fixed length. This is a simple way to perform feature extraction to represent a phenomenon in a vector space. In this paper, trace segmentation has been selected. This procedure is detailed in the next section. Then, a set of pairs (feature vectors and task label) are used to build a model of classification between tasks. In this paper, three models are explored and presented in the sections: decision tree-based classifier, multilayer perceptron neural network, and long short-term memory recurrent neural network.

### 2.7.1 Trace segmentation

Trace segmentation or variable frame rate coding (Cabral and Tattersall 1995) is a preprocessing technique that converts sequences of variable length to a constant size  $k$ . Thus, this technique is useful for algorithms that require a fixed input size such as multilayer perceptron neural networks (MLPs). It compresses consecutive points in the scanpath which are close to each other, if any, into one single point.

First and foremost, it projects the trace from 2 to 1D to stretch the initial and final points of the trace. Therefore, the size of the 1D trace equals the total length of the original 2D trace.

Then,  $k$  equidistant points are set over the trace, each at a constant distance  $s$  from its neighboring new points. The trace is filled with these new points so that the first of them is in the same position as the initial point of the original trace and the last is in the same position as the last point of the trace. Every other point is likely to fall between two separated points of the original trace.

Finally, the new points must be projected back to the original 2D space. The first and the last of the  $k$  equidistant points take the original coordinates of the first and last points of the trace. The coordinates of the others are calculated using the coordinates of the two neighboring points in the original trace using linear interpolation.

It is important to note that several consecutive points that are close to each other are more likely to be compressed to a single point. However, if two consecutive points are significantly far apart (i.e., their distance is bigger than  $s$ ) as a linear interpolation is performed, new points in the straight line between them can be used in the final trace. The procedure is summarized in Fig. 8.

### 2.7.2 Algorithm selection rationale

The goal of this study is not to present state-of-the-art results in signal processing, but to evaluate the feasibility of some ML techniques that could fit the needs of the task, whose proper resolution is relevant for the psychology field in on itself.

Therefore, each one of the models is chosen to fulfill some particular experimental purpose. XGBoost is a popular machine learning algorithm unrelated to neural networks that, therefore, could add valuable idiosyncrasy to the comparison. LSTM is a neural network technique that benefits from temporal information, while MLP does not explicitly. Thus, this comparison would serve to probe the value of temporal information in these experiments.

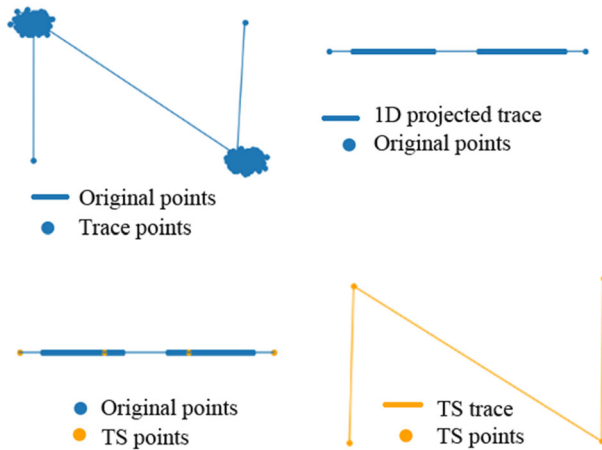


Fig. 8 Trace segmentation

### 2.7.3 Decision tree-based classifier

XGBoost (Chen and Guestrin 2016) with random forests (Liaw and Wiener 2002) provides a good trade-off between bias and variance in training. Random forest is a machine learning (ML) algorithm that, by virtue of combining decision trees, is known to be an improvement over them. Decision trees are algorithms with a high degree of variance, so they are combined by means of bagging to reduce it.

On the other hand, XGBoost is a boosting algorithm and decreases the bias and the variance of the classifiers it is used with, usually decision trees or random forests. It does this by creating a sequence of classifier instances in which each one is trained to correctly classify the mistakes of the previous instance.

In this paper, the model combines both approaches and consists of a boosted sequence of random forests.

### 2.7.4 Multilayer perceptron neural network

The multilayer perceptron (MLP) (Rosenblatt 1958) is one of the simplest neural networks as it is built out of fully connected layers of neurons and their respective nonlinear functions. In this task, MLP takes as an input the sequence with shape  $(k, 2)$  and is trained to learn the probability that it belongs to a given class. When a new unknown sequence is presented to the network, the larger output is interpreted as the highest probability and it is taken as the predicted class. Its generalization capacity had to be offset with its prediction accuracy, so a drop-out probability for each layer was established accordingly. This drop-out probability represents the chance with which a neuron may change its value to a random one. That compels the network to find weight configurations that allow for this variability and, thus, reduces the chance of overfitting the data.

However, MLP networks have a downside: They do not leverage the possible temporal correlation in the data. In this task, temporal information is strongly relevant, as the scanpath is actually defined by the order of its points. Therefore, networks that use temporal information should out-perform MLP. Such networks include RNNs, GRUs, or LSTMs. In particular, LSTMs have been used in this work.

### 2.7.5 Long short-term memory recurrent neural network

Long short-term memory recurrent neural networks (LSTM-RNNs) (Hochreiter & Schmidhuber 1997), like all RNNs, were designed to leverage the existing temporal information to deliver more grounded predictions. LSTM is an improvement over RNNs because they keep their internal state better than them, so previous information could be stored more effectively for later inferences. This is achieved by reducing the number of nonlinear operations the inner state goes through when processing a sequence.

Each coordinate is processed separately, combined with the inner state of the LSTM and the previous output to generate the output at the current step and, also, the new inner state. Intermediate outputs are ignored, and only the last is used to predict the sequence class. Likewise, dropout can be applied to the inner state of the LSTM to increase generalization.

In principle, an LSTM does not require the usage of trace segmentation as it accepts sequences of variable size. However, in practice, long sequences increase the training time. In our experiments, the sequences of thousands of coordinates are summarized using trace segmentation with a lower level of compression compared to XGBoost or MLP.

## 3 Results

Due to the small number of observations (30 individuals  $\times$  6 tasks = 180 samples), the dataset is randomly split 30 times by resampling without repetition. The classification accuracy is averaged over the 30 partitions in order to reduce the possible bias that the random partition of the participants could induce.

In each split, all the scanpaths of an individual go exclusively for training, validation, or testing. This means that a participant that has been selected to train the model is not used neither in validation nor in test.

In one split, 70% of the observations are used to train the models (21 individuals, 126 samples).

With the models based on neural networks, the individuals that are not used for training (the remaining 30%) are separated into a validation partition (4 individuals, 24 samples) and a test partition (5 individuals, 30 samples). The validation partition is used to set some parameters of the neural network, like the learning stop criterion. The test partition is used to evaluate the model performance with participants unseen during training nor validation.

With the other classifiers, the remaining 30% (9 individuals, 54 samples) unused during training goes entirely to the test partition.

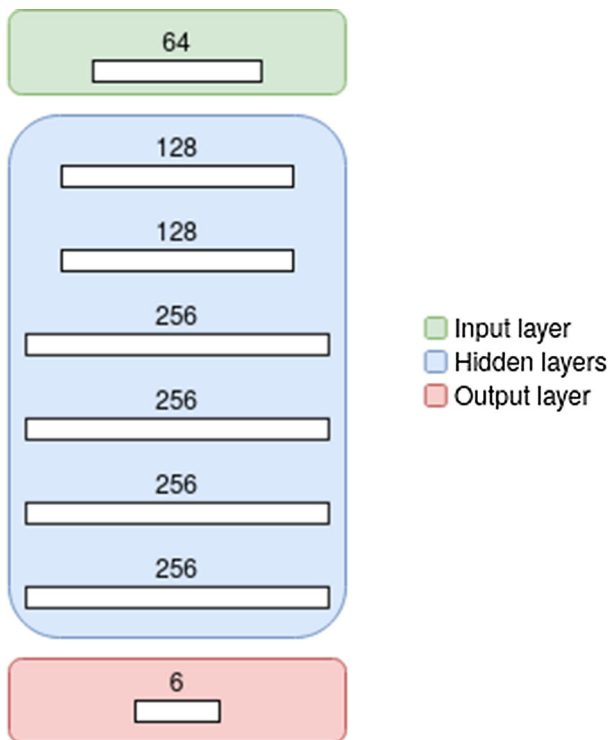
Trace segmentation was set to reduce the original trace size to 64 points in XGBoost and MLP. In the case of LSTM, the trace is compressed to 128 points. For each classifier, three trace sizes were explored (64, 128, or 256), and the one achieving the best results was retained.

LSTM and MLP have the disadvantage that their weights are randomly initialized. This initialization could be decisive in the performance of a network. To reduce the influence of a poor initialization, a poll of 5 independently trained models was built and the results were combined to select the most voted class.

Dropout and L2-norm was included in MLP and LSTM networks to increase regularization. The drop-out value was set to 25% chance to randomize a neuron for both networks, and the L2-norm and  $\beta$  value were set to  $5e-5$  and  $3e-5$ , respectively.

Adam optimizer was used to train both networks instead of plain stochastic gradient descent (SGD), as it has been observed to converge faster. The learning rate  $\alpha$  was set to  $1e-4$ , and the batch size was set to 10 samples for both networks.

The structure of the MLP network is shown in Fig. 9. It has 6 hidden layers which increase the dimensionality of the features extracted until the last layer. At the end



**Fig. 9** Structure of the MLP network. Input layer has 64 inputs, one for each feature of a scanpath. Output layer has 6 outputs, one for each of the 6 tasks to discriminate. Each output is the probability that the input layer belongs to the corresponding task

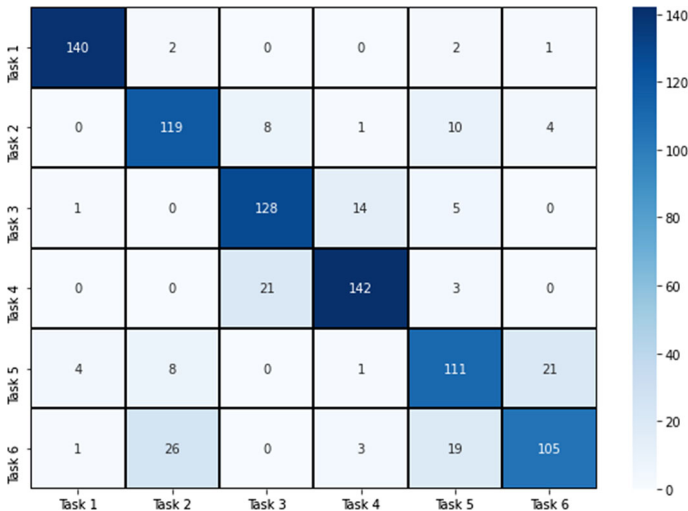
**Table 1** Accuracy for each classifier

Classifier	XGBoost RF	MLP	LSTM
Mean per-class accuracy	82.9	83.6	<b>95.7</b>
Accuracy task 1	96.6	<b>100</b>	<b>100</b>
Accuracy task 2	83.8	69.6	<b>100</b>
Accuracy task 3	86.5	93.8	<b>97.9</b>
Accuracy task 4	85.5	<b>96.8</b>	94.9
Accuracy task 5	76.6	67.6	<b>86.7</b>
Accuracy task 6	68.2	74	<b>94.6</b>

The first row is the average per-class accuracy. Next rows detail the accuracy achieved per task. All values are percentages. A bolded value highlights the model achieving the best accuracy for a specific task

of each layer, there is a RELU activation function followed by a batch normalization layer.

As shown in Table 1, LSTM achieves the highest performance, that is, a 95.7% accuracy. This is probably due to their usage of temporal information to perform the predictions. It has a ~ 10% better accuracy than MLP, with an 84.1% accuracy. Although random forest with XGBoost was the fastest to converge, it achieved the worst results with an 82.8% accuracy. Confusion matrices for random forest, MLP, and LSTM are shown in Figs. 10, 11 and 12.



**Fig. 10** Confusion matrix for the XGBoost experiment. Each cell represents a couple (ground truth task, predicted task) where the rows represent the ground truth and the columns the predicted task. The values inside are the number of scanpaths classified in this way. A perfect classification would exhibit only values in the diagonal and 0 elsewhere

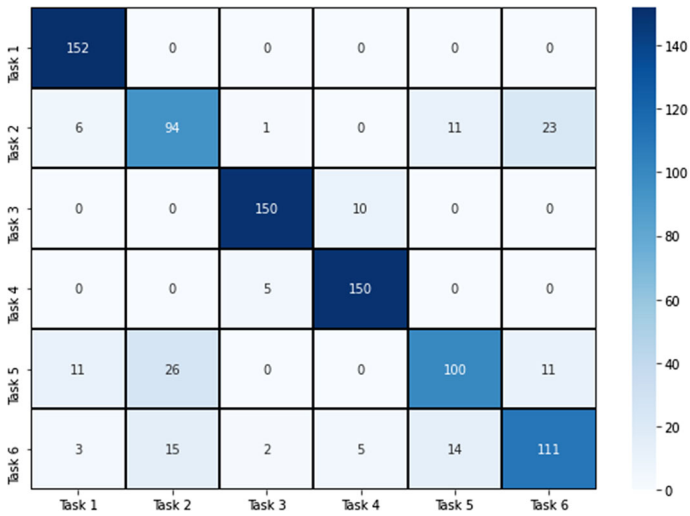


Fig. 11 Confusion matrix for the MLP experiment

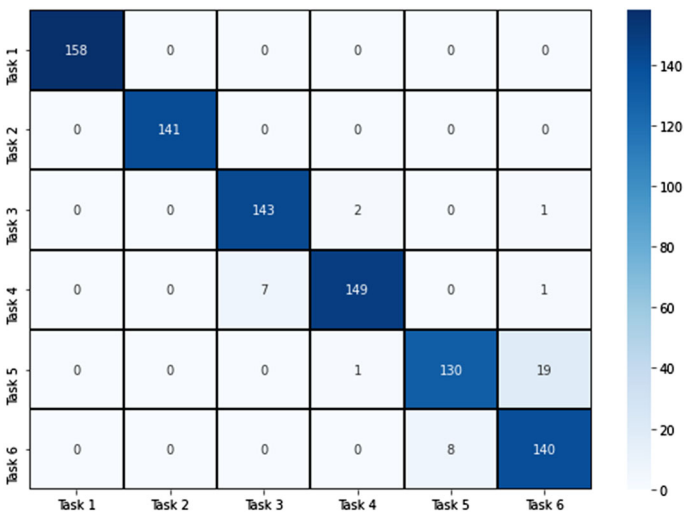


Fig. 12 Confusion matrix for the LSTM experiment

As shown in Fig. 12, LSTM struggles to separate task 5 and task 6 classes. This is reasonable as they are similar tasks which are performed over a website structure. Likewise, task 4 and task 3, which consist of two different reading tasks with or without an image, are usually misclassified for the same reason.

## 4 Discussion

### 4.1 Main findings

The accuracies achieved by all models are good enough to suggest that choices made for the data preprocessing are adequate. They validate both the use of normalized and relative coordinates (delta transformation) instead of absolute ones and the trace segmentation to represent the gaze path using a reduced number of points.

None of the algorithms required the maximum trace length (256 here) to obtain the best classification rates. This result suggests that the spatial information of the scanpaths is of sufficient quality to solve the classification task.

The three classifiers evaluated in this study were capable of discriminating between the different eye-tracking tasks based on the users' scanpaths. The classifier that achieved the best results was LSTM, with a 95.7% accuracy. XGBoost and MLP also achieved promising results and required less time to converge.

LSTM incorporates temporal information which could explain why it exhibits better performance. Moreover, LSTM uses a less compressed data (a higher  $k$  in the trace segmentation than XGBoost or MLP), which kept more information of the original trace. This could be another source of substantial improvement in the classification accuracy.

All classifiers tend to confuse task 3 with task 4. This is an expected outcome as both are reading tasks, even though they occur in different contexts (text and text plus image). The same interpretation applies to the confusion between task 5 and task 6, which are searching-for-information tasks in webpages.

XGBoost and MLP also tend to confuse task 2 (attentional look for differences) with task 5 and task 6. The scanpaths produced by these tasks contain similar large saccades that could be the origin of the confusion.

LSTM only had minor difficulties when trying to separate classes that are similar in their structure, for example, task 5 and task 6 classes, which are both tasks that included reading information from a website. It does not confuse between task 2 and tasks 5 and 6, which is an important finding.

In the current state of development, it would be possible to use LSTM to reduce the time spent in classifying the traces into their respective classes. Even if it does not achieve maximum accuracy, the amount of manual reviewing can be substantially lowered and focused to the most uncertain classifications.

Moreover, this model can also be used within an incremental learning strategy. In this strategy, a model is initially trained with a subset of the whole dataset and the rest of the dataset is labeled using this model. The whole dataset just needs to be reviewed, but not classified, to correct possible errors. The same strategy can be applied to any new subject: The model proposes a classification associated to a confidence level that could be reviewed to confirm or reject the hypothesis. This makes it possible to iteratively improve the model by specifying the errors.

## 4.2 Potential applications

In this work, our algorithm analyzes complete tasks and therefore its current use would be limited to distinguishing between the different tasks proposed at the generic level. However, we would like to propose for future works some examples of potential uses of the study findings in different contexts. As we mentioned earlier in the beginning of the text, task performance is one of the most widely used metrics in usability studies (Tullis and Bill 2013). Thus, our first proposal is directly related to facilitating a new quantitative tool for the UX expert work. Eye-tracking is a widely used technique to analyze user behavior with technology. While acknowledging this, we also argued that the default analytical options offered by commercial software tools are insufficient, which has forced UX researchers to try to analyze the information in other ways, often manually. A manual, qualitative analysis is not only time-consuming, but also introduces potential biases (Norgaard and Hornbaek 2006). The use of quantitative techniques, such as those applied in the present study, is more robust to error, but often requires time and the involvement of qualified professionals to extract, prepare, and analyze the data. Because scanpath patterns can reveal interesting information for research, clinical, and commercial purposes (e.g., during attentional or reading tasks), it would be highly desirable to develop analytical procedures that provide ready-to-interpret scanpath outcomes from eye-tracking systems, similar to other data processed by commercial software such as AOI clusters. For example, we know that all users have the same descending zigzag pattern when reading (see Fig. 5). Thus, the researcher could select and set up the reading areas of a website like an “AOI-reading” or mark two objects like a couple of “AOI-comparison.” Then, the software analysis could return an easy-to-interpret parameter from 0 to 100, indicating how closely the collected behavior matched the expected behavior. This would allow, for example, to analyze hours of interaction in a minute and to quickly identify the parts of the web which are not performing as expected with a quantitative parameter. These desirable features would not only save researchers a lot of time, but would also advance our knowledge of user behavior with ICTs.

On the other hand, the designed tasks for this study (read, attentional tasks, and free browsing) represent some behaviors that are present in many web platforms for commercial purposes, in online teaching platforms, or in online psychological treatment programs. Thus, the benefits of this work can be multiple. For example, being able to distinguish between a comparison pattern and a reading pattern could allow us to know which elements of a product are key to choose it, could help us adapt advertising in real time to the type of consumer, or could inform us of the need to provide more elements of visual comparison or text information in function of web behavior.

The present study findings could also be very helpful to apply in other contexts. For example, in an online learning platform, it would be possible to measure the percentage of content that has been truly read, the reading speed, and information to adapt the exercises that are presented to the reading characteristics of each student. Another example in the mental health context, item comparison tasks as well as reading tasks are frequent items in cognitive rehabilitation (Cicerone et al. 2019). For users with cognitive impairment, task performance may vary from one day to the next or even



oscillate within the same day. Thus, providing exercises that match with the cognitive level of the user is essential, since providing tasks that are too easy would not stimulate the user sufficiently and selecting tasks that are too difficult would be too demanding and cause frustration. So, we propose a system provided with an eye-tracker that would react to the user performance in reading or attentional tasks in real time and would rapidly adapt the cognitive exercises to the user's performance level.

In summary, this knowledge could allow us to improve the way we design or adapt the interfaces to the user's behavior and could help us enhance the way the user interacts with shopping applications, online learning, or telepsychology, to give a few examples.

### 4.3 Limitations and future directions

One of the most important limitations is related to the nature of this study, given that the tasks performed represent behaviors in laboratory conditions. For this reason, it would be an important future research to replicate this study in real-use conditions, such as comparison of products in e-commerce, students' performance in e-learning platforms, or the design of cognitive rehabilitation systems that adapt the training program in real time according to the user's performance.

This work also presents additional limitations regarding the number of subjects and data inputs used to train the classifiers.

The encouraging results, however, show potential toward exploring this line of research. In the future, we aim to increase the data set with more participants to provide more generalizable results. In addition, data augmentation techniques could be used to improve classification accuracy. With a larger dataset, additional evaluations could be conducted in the future to improve the ML model. This would also allow us to train more complex models with additional layers in the neural network. On the one hand, LSTM with trace segmentation yielded very promising results, but imposed several restrictions on the data. Comparative evaluations could be conducted with LSTM without trace segmentation to assess whether there are situations in which this approach could be more effective. On the other hand, a study could be conducted regarding how different variations of the trace segmentation length affect the accuracy of the model.

This paper proposes the use of the LSTM classifier to differentiate between several web tasks. While these tasks are important, however, they do not represent the global behavior of users. More research is needed on the relationship between user behavior and the scanpaths generated during the use of ICTs.

In the future, we aim to work toward standardizing, using a simple value (0–100%), the level of coincidence between the expected behavior and the one found by the classifier (e.g., percentage of reading behavior in a specific area marked as AOI-reading). This model would be helpful to categorize the performance of participants during the proposed web tasks, thus potentially identifying abnormal behaviors that could be representative of relevant mental states. For example, specific scanpath patterns

could be representative to identify a given disorder, such as generalized anxiety disorder, major depressive disorder, attention-deficit/hyperactivity disorder, or autism spectrum disorder.

## 5 Conclusions

The aim of this work was to propose a new quantitative way to analyze the web user behaviors through eye-tracking data analyses with machine learning. More specifically, we analyze the scanpaths of 6 typical web tasks, which encompass reading, free browsing, attentional tasks, and item comparisons.. The results showed that it is possible to classify and distinguish between the 6 different web tasks proposed in this study based on the user's scanpath with a 82.8–95.6% accuracy. The different ML classifiers evaluated in this manuscript have shown to be promising to provide reliable information based on the user's eye-tracking data. In the discussion, the authors propose different ways to use this finding in a practical manner, either as a new quantitative parameter for task analysis or as a fundamental part of new systems that are able to adapt to the user's performance in real time. The actual implementation in these scenarios, however, is out of the scope of this work.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results. This publication has been co-financed by IVACE through the nominative line to support the non-economic activity of ITI during the year 2021, with file number IMAMCN/2021/1.

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Asan, O., Yang, Y.: Using eye trackers for usability evaluation of health information technology: a systematic literature review. *JMIR Hum. Factors* 2(1), 1–14 (2015). <https://doi.org/10.2196/humanfactors.4062>
- Beesley, T., Pearson, D., Le Pelley, M.: Eye tracking as a tool for examining cognitive processes. In *Biophysical measurement in experimental social science research*, (2019). <https://doi.org/10.1016/b978-0-12-813092-6.00002-2>

- Berger, C., Winkels, M., Lischke, A., Höppner, J.: GazeAlyze: a MATLAB toolbox for the analysis of eye movement data. *Behav. Res. Methods* **44**(2), 404–419 (2012). <https://doi.org/10.3758/s13428-011-0149-x>
- Borji, A., Itti, L.: Defending yarbus: eye movements reveal observers' task. *J. vis.* **14**(3), 1–22 (2014). <https://doi.org/10.1167/14.3.29>
- Bott, N.T., Madero, E.N., Glenn, J.M., Lange, A.R., Anderson, J.J., Newton, D.O., Brennan, A.H., Buffalo, E.A., Rentz, D.M., Zola, S.M.: Device-embedded cameras for eye tracking-based cognitive assessment: validation with paper-pencil and computerized cognitive composites. *J. Med. Internet Res.* **20**(7), e11143 (2018)
- Buscher, G., Dumais, S. T., Cutrell, E.: The good, the bad, and the random: an eye-tracking study of ad quality in web search. In: Proceedings of the 33<sup>rd</sup> international ACM SIGIR conference on Research and development in information retrieval, pp. 42–49. (2010). <https://doi.org/10.1145/1835449.1835459>
- Cabral, E. F., Tattersall, G. D.: Trace-segmentation of isolated utterances for speech recognition. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing—Proceedings, vol. 1, pp. 365–368. (1995). <https://doi.org/10.1109/icassp.1995.479597>
- Castelhano, M.S., Henderson, J.M.: Stable individual differences across images in human saccadic eye movements. *Can. J. Exp. Psychology/revue Can. Psychol. Exp.* **62**(1), 1–14 (2008). <https://doi.org/10.1037/1196-1961.62.1.1>
- Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22<sup>nd</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. (2016). <https://doi.org/10.1145/2939672.2939785>
- Cicerone, K.D., Goldin, Y., Ganci, K., Rosenbaum, A., Wethe, J.V., Langenbahn, D.M., Malec, J.F., Bergquist, T.F., Kingsley, K., Nagele, D., Trexler, L., Fraas, M., Bogdanova, Y., Harley, J.P.: Evidence-based cognitive rehabilitation: systematic review of the literature from 2009 through 2014. *Arch. Phys. Med. Rehabil.* **100**(8), 1515–1533 (2019). <https://doi.org/10.1016/j.apmr.2019.02.011>
- Coco, M.I., Keller, F.: Classification of visual and linguistic tasks using eye-movement features. *J. vis.* **14**(3), 1–18 (2014). <https://doi.org/10.1167/14.3.11>
- Colombo, C., Comanducci, D., Bimbo, A.D.: Robust tracking and remapping of eye appearance with passive computer vision. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **10**(1145/1314303), 1314305 (2007)
- Crosby, M. E., Peterson, W. W.: Using eye movements to classify search strategies. In: Proceedings of the human factors society. (1991). <https://doi.org/10.1177/154193129103502012>
- Djamasbi, S.: Eye tracking and web experience. *AIS Trans. Hum. Comput. Interact.* **6**(2), 37–54 (2014)
- Djamasbi, S., Siegel, M., Tullis, T.: Generation Y, web design, and eye tracking. *Int. J. Hum. Comput. Stud.* **68**(5), 307–323 (2010). <https://doi.org/10.1016/j.ijhcs.2009.12.006>
- Dolezalova, J., Popelka, S.: Scangraph: a novel scanpath comparison method using visualisation of graph cliques. *J. Eye Mov. Res.* **9**(4), 1–13 (2016)
- Drusch, G., Bastien, J. C., Paris, S.: Analysing eye-tracking data: from scanpaths and heatmaps to the dynamic visualisation of areas of interest. In: International Conference on Applied Human Factors and Ergonomics, 2014, Krakow, Poland. (2014). Retrieved from: <https://hal.archives-ouvertes.fr/hal-01223743/document>
- Duchowsky, A.: Eye tracking methodology: theory and practice. Springer, Berlin (2003)
- Eckstein, M.K., Guerra-Carrillo, B., Miller Singley, A.T., Bunge, S.A.: Beyond eye gaze: What else can eyetracking reveal about cognition and cognitive development? *Dev. Cogn. Neurosci.* **25**, 69–91 (2017)
- Eraslan, S., Yesilada, Y., Harper, S.: Eye tracking scanpath analysis techniques on web pages: a survey, evaluation and comparison. *J. Eye Mov. Res.* **9**(1), 1–19 (2016a)
- Eraslan, S., Yesilada, Y., Harper, S.: Scanpath trend analysis on web pages: clustering eye tracking scanpaths. *ACM Trans. Web* **10**(4), 1–35 (2016b). <https://doi.org/10.1145/2970818>
- Eraslan, S., Yaneva, V., Yesilada, Y., Harper, S.: Web users with autism: eye tracking evidence for differences. *Behav. Inf. Technol.* **38**(7), 678–700 (2019). <https://doi.org/10.1080/0144929X.2018.1551933>
- Geller, J., Winn, M.B., Mahr, T., Mirman, D.: GazeR: a package for processing gaze position and pupil size data. *Behav. Res. Methods* (2020). <https://doi.org/10.3758/s13428-020-01374-8>
- Ghasemifard, N., Shamsi, M., Kenari, A.R.R., Ahmadi, V.: A new view at usability test methods of interfaces for human computer interaction. *Global J. Comput. Sci. Technol. Hardw. Comput.* **15**(1), 1–9 (2015)
- Greene, M.R., Liu, T., Wolfe, J.M.: Reconsidering yarbus: a failure to predict observers' task from eye movement patterns. *Vis. Res.* **62**, 1–8 (2012). <https://doi.org/10.1016/j.visres.2012.03.019>

- Gual-Montolio, P., Jaén, I., Martínez-Borba, V., Castilla, D., Suso-Ribera, C.: Using artificial intelligence to enhance ongoing psychological interventions for emotional problems in real- or close to real-time: a systematic review. *Int. J. Environ. Res. Public Health* **19**(13), 7737 (2022). <https://doi.org/10.3390/ijerph19137737>
- Haass, M. J., Matzen, L. E., Butler, K. M., Armenta, M.: A new method for categorizing scanpaths from eye tracking data. In: *Eye tracking research and applications symposium (ETRA)*, (2016). <https://doi.org/10.1145/2857491.2857503>
- Hackos, J.T., Redish, J.C.: *User and task analysis for interface design*. Wiley, New York (1998)
- Haji-Abolhassani, A., Clark, J.J.: An inverse yarbus process: predicting observers' task from eye movement patterns. *Vis. Res.* **103**, 127–142 (2014). <https://doi.org/10.1016/j.visres.2014.08.014>
- Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hooge, I.T., Holleman, G.A., Haukes, N.C., Hessels, R.S.: Gaze tracking accuracy in humans: one eye is sometimes better than two. *Behav. Res. Methods* **51**(6), 2712–2721 (2019). <https://doi.org/10.3758/s13428-018-1135-3>
- Ioannou, C., Nurdiani, I., Burattin, A., Weber, B.: Mining reading patterns from eye-tracking data: method and demonstration. *Softw. Syst. Model.* **19**(2), 345–369 (2020). <https://doi.org/10.1007/s10270-019-00759-4>
- Kanan, C., Ray, N. A., Bseiso, D. N. F., Hsiao, J. H., Cottrell, G. W.: Predicting an observer's task using multi-fixation pattern analysis. In: *Eye Tracking Research and Applications Symposium (ETRA)*, (2014). <https://doi.org/10.1145/2578153.2578208>
- Kao, W.C., Shen, C.W., Wu, C.J.: Reading behavior analysis with gaze tracking data. In: *IEEE International conference on consumer electronics–Taiwan*, pp. 1–2, (2018)
- Kar, A., Corcoran, P.: GazeVisual: a practical software tool and web application for performance evaluation of eye tracking systems. *IEEE Trans. Consum. Electron.* (2019). <https://doi.org/10.1109/TCE.2019.2912802>
- Kardan, O., Berman, M.G., Yourganov, G., Schmidt, J., Henderson, J.M.: Classifying mental states from eye movements during scene viewing. *J. Exp. Psychol. Hum. Percept. Perform.* **41**(6), 1502–1514 (2015). <https://doi.org/10.1037/a0039673>
- Kardan, O., Yourganov, G., Henderson, J.M., Berman, M.G.: Observers' cognitive states modulate how visual inputs relate to gaze control. *J. Exp. Psychol. Hum. Percept. Perform.* **42**(9), 1429–1442 (2016). <https://doi.org/10.1037/xhp0000224>
- Król, M.E., Król, M.: The right look for the job: decoding cognitive processes involved in the task from spatial eye-movement patterns. *Psychol. Res.* **84**(1), 245–258 (2020). <https://doi.org/10.1007/s00426-018-0996-5>
- Kumar, A., Tyagi, A., Burch, M., Weiskopf, D., Mueller, K.: Task classification model for visual fixation, exploration, and search. In: *Eye tracking research and applications symposium (ETRA)*. (2019). <https://doi.org/10.1145/3314111.3323073>
- Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(9), 1465–1479 (2006). <https://doi.org/10.1109/TPAMI.2006.188>
- Liaw, A., Wiener, M.: Classification and regression by randomForest. *R News* **2**(3), 18–22 (2002)
- Mele, M.L., Federici, S.: A psychotechnological review on eye-tracking systems: towards user experience. *Disabil. Rehabil. Assist. Technol.* **7**(4), 261–281 (2012). <https://doi.org/10.3109/17483107.2011.635326>
- Norgaard, M., Hornbaek, K.: What do usability evaluators do in practice? An explorative study of think-aloud testing. In: *Proceedings of designing interactive systems*, pp. 209–218. University Park, PA, (2006)
- Papavlasopoulou, S., Sharma, K., Giannakos, M.N.: Coding activities for children: coupling eye-tracking with qualitative data to investigate gender differences. *Comput. Hum. Behav.* **105**(7491), 105939 (2020). <https://doi.org/10.1016/j.chb.2019.03.003>
- Pfeiffer, J., Pfeiffer, T., Meißner, M., Weiß, E.: Eye-tracking-based classification of information search behavior using machine learning: evidence from experiments in physical shops and virtual reality shopping environments. *Inf. Syst. Res.* **31**(3), 675–691 (2020). <https://doi.org/10.1287/isre.2019.0907>
- Raptis, G. E., Katsini, C., Avouris, N., Belk, M., Fidas, C., Samaras, G.: Using eye gaze data & visual activities to infer human cognitive styles: Method & feasibility studies. In: *UMAP 2017–Proceedings of the 25th conference on user modeling, adaptation and personalization*. (2017). <https://doi.org/10.1145/3079628.3079690>

- Roa-Martinez, S., Vidotti, S.: Eye tracking y usabilidad en ambientes informacionales digitales: revisión teórica y propuesta de procedimiento de evaluación. *Transinformação* (2020). <https://doi.org/10.1590/1678-9865202032e190067>
- Rosch, J.L., Vogel-Walcutt, J.J.: A review of eye-tracking applications as tools for training. *Cogn. Technol. Work* **15**(3), 313–327 (2013). <https://doi.org/10.1007/s10111-012-0234-7>
- Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **65**(6), 386–408 (1958). <https://doi.org/10.1037/h0042519>
- Sauro, J., Lewis, J.R.: Quantifying the user experience: practical statistics for user research. M. Kaufmann, Waltham (2016)
- Schall, A., Bergstrom, J.: Eye tracking in user experience design. In: Bergstrom, J.E., Schall, A. (eds.) *Introduction to eye tracking*, pp. 3–26. Elsevier, Netherlands (2014)
- Singh, H., Singh, J.: Human eye tracking and related issues: a review. *Int. J. Sci. Res. Publ.* **2**(9), 1–9 (2012)
- Tang, S., Reilly, R.G., Vorstius, C.: EyeMap: a software system for visualizing and analyzing eye movement data in reading. *Behav. Res. Methods* (2012). <https://doi.org/10.3758/s13428-011-0156-y>
- Tobii TX 300. (2015). Tobii TX300 Eye tracker product description. Tobii Technology AB
- Tsai, M.J., Hsu, P.F., Pai, H.T.: Eye-tracking data analyzer (EDA) Developed For Educational Researchers: A Sample Module of LSA. In: *Proceedings–2018 international symposium on educational technology, ISET 2018*. (2018). <https://doi.org/10.1109/ISET.2018.00050>
- Tullis, T., Bill, A.: *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Morgan Kaufmann, Waltham, MA (2013)
- Urruty, T., Lew, S., Ihadadene, N., Simovici, D.A.N.A.: Detecting eye fixations by projection clustering. *ACM Trans. Multimed. Comput. Commun. Appl.* **3**(4), 1–20 (2007). <https://doi.org/10.1145/1314303.1314308>
- Vidal, M., Bulling, A., Gellersen, H.: Detection of smooth pursuits using eye movement shape features. In: *Proceedings of the Symposium on Eye Tracking Research and Applications–ETRA '14*. (2012). <https://doi.org/10.1145/2168556.2168586>
- Voßkühler, A., Nordmeier, V., Kuchinke, L., Jacobs, A.M.: OGAMA (open gaze and mouse analyzer): open-source software designed to analyze eye and mouse movements in slideshow study designs. *Behav. Res. Methods* **40**(4), 1150–1162 (2008). <https://doi.org/10.3758/BRM.40.4.1150>
- Yarbus, A.: *Eye movements and vision*. Plenum Press, New York (1967)
- Zaki, T., Islam, M.N.: Neurological and physiological measures to evaluate the usability and user-experience (UX) of information systems: a systematic literature review. *Comput. Sci. Rev.* **40**, 100375 (2021). <https://doi.org/10.1016/j.cosrev.2021.100375>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Diana Castilla** is a psychologist with an interest in the study of human–computer interaction and its application in the design of mental health applications; she is professor in the University of Valencia, and director of SAEM-TIC, Research group in mental health technologies at University of Valencia..

**Omar Del Tejo Catalá** is a computer science engineer with an interest in deep learning techniques applied to several fields such as object detection, medical applications, reinforcement learning, and image classification; he is researcher in the Instituto Tecnológico de Informática (ITI) of Universitat Politècnica de València.

**Patricia Pons** is a computer science engineer with an interest in immersive technologies and mixed reality, technology to support mental health and well-being, and natural user interfaces; she is researcher in the Instituto Tecnológico de Informática (ITI) of Universitat Politècnica de València.

**François Signol** is a telecommunication engineer with an interest in bioinformatics and how bringing artificial intelligence and machine learning techniques to the services of health; he is researcher and project manager in the Instituto Tecnológico de Informática (ITI) of Universitat Politècnica de València.

**Beatriz Rey** is a telecommunication engineer with an interest in health applications of different technological and computer-based tools, such as virtual reality, digital signal processing, and digital image processing; she is associate professor in the Graphic Engineering department of the Universitat Politècnica de València.

**Carlos Suso-Ribera** is a psychologist with an interest in the contribution of psychosocial factors in the evolution of patients with chronic pain, as well as in the benefits of new technologies in their assessment and treatment; he is associate professor in the Universitat Jaume I.

**Juan-Carlos Perez-Cortes** is a computer science engineer with an interest in medical imaging, industrial software, computer vision, pattern recognition, and free software; he is Full Professor in the Polytechnic University of Valencia and director of PRAIA (Perception Recognition Learning and Artificial Intelligence) research group at Instituto Tecnológico de Informática (ITI).