Detecting Emotions Through Machine Learning for Automatic UX Evaluation

Giuseppe Desolda, Andrea Esposito, Rosa Lanzilotti, Maria F. Costabile

Detecting Emotions Through Machine Learning for Automatic UX Evaluation

Giuseppe Desolda, Andrea Esposito, Rosa Lanzilotti, Maria F. Costabile

Computer Science Department, University of Bari Aldo Moro, Italy

{giuseppe.desolda, andrea.esposito, rosa.lanzilotti, maria.costabile}@uniba.it

Abstract. Although User eXperience (UX) is widely acknowledged as an important aspect of software products, its evaluation is often neglected during the development of most software products, primarily because developers think that it is resource-demanding and complain about the fact that is scarcely automated. Various attempts have been made to develop tools that support and automate the execution of tests with users. This paper is about an ongoing research work that exploits Machine Learning (ML) for automatic UX evaluation, specifically for understanding users' emotions by analyzing the log data of the users' interactions with websites. The approach described aims at overcoming some limitations of existing proposals based on ML.

Keywords: Usability, User eXperience, automatic UX evaluation.

1 Introduction

User eXperience (UX) has become an increasingly important aspect of software products. It extends the more traditional quality of usability, focused primarily on ease of learning and ease of use. Standard ISO 9241-11:1988 defines UX as *a person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service*. Other definitions are available in the literature (e.g., [15]). Designing for UX is much more than designing for the traditional attributes of usability. UX also refers to attributes related to people feelings and emotions, such as pleasure, fun, surprise, intimacy, joy; it focuses on beautiful (harmonious, clear), emotional (affectionate, lovable), stimulating (intellectual, motivational), and also on tactile (smooth, soft), acoustic (rhythmic, melodious).

Some main reasons why usability and UX are still neglected during the development of most software products is that developers think that usability and UX evaluations are resource-demanding and require specific expertise because they are scarcely automated [19, 31]. In order to foster more attention on usability and UX, the automation of evaluation tests, performed remotely, could be a solution due to its great potential of reducing costs. Remote usability testing was defined around the mid-'90s to limit some usability testing drawbacks [21]; it refers to user testing performed by evaluators who are in different physical locations from the participants and might operate at different times [10]. Software tools for remote testing were developed to allow participants to test systems from their locations, at times when it is convenient for them. The tools automatically gathered and stored data about the tests. Examples of more recent tools are Userlytics [27], Loop11 [16], eGLU-Box PA [9]; while complete automation is still far from possible, these tools take great advantage of current technology and provide more useful features with respect to the tools of the '90s. For example, eGLU-Box PA also allows the detection of issues concerning UX, called “UX smells” (in a similar fashion to the commonly known expression “code smells”), using visualization techniques that show the paths followed by participants to carry out tasks on websites during a test [4].

Since emotions are important elements of UX, some authors are looking for ways of identifying users’ emotions by analyzing the users’ interaction with the system (e.g., mouse movements) through machine learning (ML) algorithms [3, 5, 12, 20, 29, 32]. These proposals are not mature enough due to several limitations, such as 1) the proposed models are often trained on datasets that are too small, contain data collected in controlled environments (thus with low ecological validity), and the emotions are self-reported by the users; 2) there is not a comparison of different ML algorithms and no clear indication on the most suitable one for this kind of task is provided; 3) the prediction provided by the ML algorithm is always in relation to long timespan, thus making impossible to understand users’ emotions during intermediate moments.

This paper presents an ongoing research work that exploits ML for emotion detection, with the aim to overcome the above limitations. It provides the following contributions. First, a dataset of users’ emotions and interaction logs of real users that interact with real websites “in the wild” for 30 days is built. Second, the results of the comparison of the four ML algorithms mainly used for detecting emotions are provided. Third, the resulting classification models can predict users' emotions moment by moment during the users’ interaction. The paper is organized as follows. Section 2 discusses the rationale and background of our research. Section 3 illustrates how the dataset was built, and Section 4 presents the comparison of four ML algorithms to predict emotions felt by users interacting with websites. Section 5 provides the conclusions and highlights future works.

2 Background and Related Work

UX integrates the usability concept by emphasizing more subjective feelings. A valuable UX increases user satisfaction: an aesthetically pleasing interface, which is easily navigable and presents updated and trustable content, heavily increases user satisfaction [30].

Various evaluation techniques have been proposed (see, e.g., [14, 22]). One of the most successful is user testing, it is considered the most complete form of evaluation because it assesses usability and UX through samples of real users [18]. However, it is often perceived as impractical mainly because it is very resource-demanding [2]; for small companies/organizations, the cost of recruiting users and expert evaluators and transporting them to different locations can be prohibitive [26, 31]. Various approaches for semi-automating user testing have been proposed. The tools for remote user testing developed about 20 years ago (e.g., [7]) used the technology available at that time, which was limited. Technology advances in recent years have opened up new scenarios. Nowadays, it is possible to realize web applications with functions for screen recording, user-interaction tracking, access to peripheral devices, such as webcams and microphones to capture the face and speech of the participants during user tests. The tools developed in the last five years (e.g., [16, 28]) provide several features to better support automatic testing and to collect much more qualitative data that may help discover further usability issues. The main limitation is that it is impossible to keep track of the actual test reliability: users may choose to "rush" the test or may be influenced by the testing environment (this is an issue that also affects "classic" methods), thereby distorting the final results. Finally, while reducing the costs with respect to conducting the tests in-house, the costs are still higher than fully automatic evaluations.

To move toward automatic UX evaluation, some research works exploit ML algorithms to build models that, starting from interaction logs, predict the emotions felt by users. For each user interface the users interact with, or for each task the users perform, these models return the score of each emotion. However, these solutions still have room for improvement. For example, the datasets used to train the models are built during in-lab sessions but, as it is widely known, user interaction in the lab is biased [23], thus the resulting models are also biased. Other datasets are built during the execution of a limited number of tasks or on a specific model [1, 3, 5, 12, 29], thus reducing the generalizability of the predictions. In some cases, users are asked about their emotions at the end of a task or after a long time span (e.g., *n* minutes) [1, 5, 12, 20, 32] by using a Likert scale [1] or a SAM (Self-Assessment Manikin) scale [12]. This determines a strong limitation caused by the so-called *peak–end rule* [6]: it is a psychological behavior of people that judge an experience depending on how they felt during its peak (the most intense point) and at the end, rather than based on the total sum or average of every moment of the experience. Thus, these datasets cannot be representative of the overall users' emotions since users tend to recall the last part of the interaction and/or the pick. Another limitation regards the data pre-processing since datasets used until now are not further refined, excluding simple filters [29] or aggregation in time frames [1]. Finally, to the best of our knowledge, no previous works aim at predicting users' emotions at a lower granularity, i.e., at predicting emotions every second. This low-level prediction helps evaluators in understanding users' emotions not only during the visit of a webpage or during a task execution but, more deeply, it permits identifying emotions felt by users during the interaction with specific elements of the webpages. In this way, it is possible to understand, for instance, if UI elements like a menu, video, widget, etc., determine positive or negative emotions.

3 Construction of in-the-wild dataset

With the objective of overcoming some of the limitations identified in the literature, this paper presents an ongoing research work to provide researchers, practitioners, and companies with a software tool for the automatic detection of emotions. This tool, which is still under development, consists of 1) a JavaScript snippet to be integrated into the pages of a website, and 2) a dashboard to analyze users' emotions. In particular, the snippet asynchronously tracks the user's interactions (mouse movements and keyboard press) and sends them to a web server. The dashboard reports for each webpage the results of an ML model applied to the logs of all the users that visited that page, to predict their average emotions. **Fig. 1** reports an example of a prototype of the dashboard that visualizes a heatmap depicting the emotion, joy in this example, the users felt while interacting with the webpage. This heatmap indicates in red the web pages elements with high emotion values, while in blue elements with low values. On the right side, there is a legend and a radio-box menu to change the emotion visualized in the heatmap. It is worth noticing that some users might feel emotions not related to the specific website but caused by external factors (e.g., mood, tiring, etc.). However, the final values computed by the ML models and visualized on the heatmap are an average of emotions of hundreds, thousand or more users, thus possible wrong values are mitigated or cancelled by the sample size.

Fig. 1. Example of heatmap showing an emotion intensity on a webpage.

At the current stage of our research, we are focusing on two activities related to the construction of the prediction model: 1) the creation of a significant dataset overcoming the limitations identified in the literature, and 2) the comparison of different ML methods for classification. We built a new dataset according to the following requirements:

1. *Data are collected in the wild*, i.e., while users naturally interact with any website for any kind of activity, without any restriction. In this way, the models trained on these data better reflect the real user behavior and emotions.
2. *Data are collected with low granularity*, possibly every a few seconds. This low-level sampling enables the emotion prediction moment by moment, in order to understand users' emotions while interacting with specific website elements (e.g., a menu, a label) instead of more high-level predictions that simply indicate users' emotions during the interaction with a webpage or during task execution, as it occurs in [5, 29].
3. *Data are collected transparently.* Interaction tracking must be transparent to users. Similarly, users should not be asked to stop their activity to declare their emotions, as required in [1, 5, 12, 20, 32].

4. *Emotions are automatically computed by analyzing the user's facial expressions*. Facial expression analysis avoids both the use of intrusive hardware for emotion detection (e.g. brain-computer-interfaces devices [32]) and to stop users during the test to ask them to declare their emotions. In addition, this analysis is more precise than self-declared emotions, which also suffers from the peak-end rule problem.
5. *Participants' data anonymization is guaranteed*. This is required for ethical issues. It also fosters participant recruiting.

We recruited 12 volunteers of different ages (mean age 32.3) and gender (6 women). They were asked to install for 30 days on their browser (Chrome or Firefox was required) a plugin we purposely developed. This plugin records, every 2 seconds, a picture from the webcam that frames the user's face as well as mouse and keyboard logs (to avoid a keylogger effect, the plugin only records the type of key pressed, i.e., number, letter, special character). This data was anonymously sent, through an SSL connection, to a web server where a Node.js module manages them. Every time this module receives the data, it invokes the Affdex SDK [17] to translate the picture into a set of pairs [emotion-value], according to the EMFACS (Emotional Facial Action Coding System) model which identifies seven emotions: Anger, Contempt, Disgust, Joy, Fear, Sadness, and Surprise [8]. It is worth remarking that recent advances in computer visions and Deep Learning make visual emotion recognition, and in particular the Affdex tool, as reliable as human coding [24, 25], as well as precise like more advanced and invasive instruments as facial electromyography [13]. Then, the Node.js module permanently discards the picture and then associates the emotions to the user logs. The result is an *interaction object* stored in a MongoDB database that describes what the user was doing and his/her emotions in the sampling moment. At the end of the 30 days, around 3GB of data on the users' interactions on 473 websites were stored.

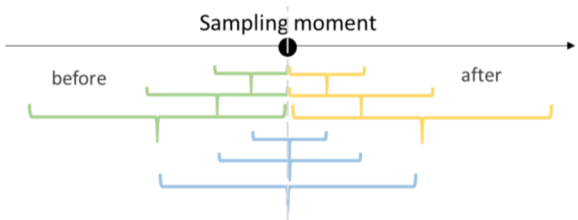


Fig. 2. Three-time windows of different lengths (25/50/100 milliseconds) were computed before (green), after (orange), and before and after (blue) each sampled moment. Nine-time windows are associated with each interaction object.

A preprocessing phase was performed before using the dataset with the ML algorithms. First, since the emotion values returned by Affdex SDK range from 0 to 100, in line with previous works we discretized such values in three classes [1, 5, 12, 20, 32], i.e., low (from 0 to 33), medium (from 34 to 66), and high (from 67 to 100). Then, each interaction object was extended by computing further metrics, namely mouse speed, acceleration, and direction. In addition, metrics measuring three-time windows of variable length (25/50/100 milliseconds) were calculated before, after, and before&after the sampling moment (see **Fig. 2**). A total of 9 windows were computed and each of them contains: average speed among all axes, average number of clicks per second, average idle time, average number of mouse movements per second, average number per second of mouse's trajectory changes, URL change rate, number of website changes, average number of key pressed per second per type. In the end, each interaction object contains the interaction log in the sampling moments, the 7 emotion values, and the 9-time windows, each one characterized by additional features. The final dataset contains 527853 interaction objects characterized by 549 features. An example of a resulting interaction object described in JSON format is reported at this link https://bit.ly/3s3BvmL.

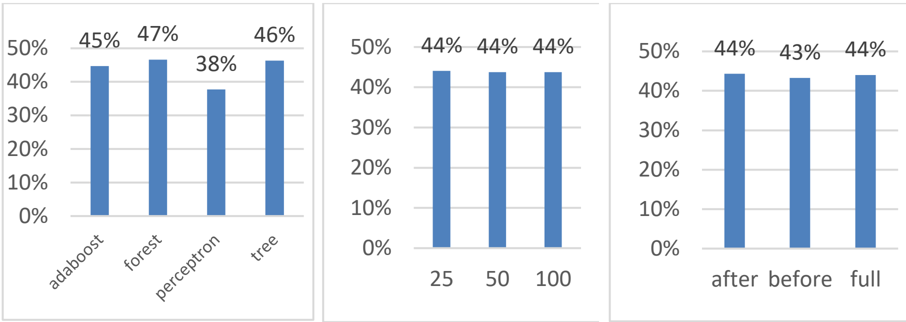
4 Comparison of learning algorithms for user's emotions classification

In the literature, there are no comparisons between ML algorithms for emotion prediction starting from users' interactions. Moreover, there is no comparison among the best sets of features (related to keyboard and mouse usage) to predict emotions. For this reason, we compared different learning algorithms, in particular, all the ones already used in previous works, i.e., binary decision trees, random forests, AdaBoost, and Multi-Layered Perceptron. To select the best set of features for each kind of model, a forward feature selection algorithm was applied.

The training phase poses an important problem related to the required computational time and resources. For each type of learning algorithm, we built 7 models (one for each emotion). For each of these 7 models, we trained 9 models, each one related to the different time windows. This led to the computation of a total of 252 models (4 learning algorithms × 7 emotions × 9-time windows). The k-fold cross-validation used to evaluate the models' performance (with k=10) and the feature selection algorithm further amplify this problem, leading to the computation of thousands of intermediate models.

The model computation were performed by writing a Python 3 script that uses *SciKit-Learn* for model computation and *MLxtend* library for the feature selection algorithm. The computation was performed on an HPC cluster machine provided by our University. This is equipped with CentOS 7.6 – 64bit, and it has up to 400 physical cores (800 considering hyperthreading), each with 4 GB of RAM. For this computation, only 8 cores were used (the allocation of more cores required too much time). To make possible the computation of all the combinations of the models in a reasonable amount of time, we empirically established that no more than 0.1% of the dataset could be used. To deal with the "class imbalance problem" [11], a stratified sampling was applied, so that each class was represented by a similar number of examples (463 objects for each class of each emotion). The entire computation lasted 35 hours and 43 minutes.

As reported in **Fig. 3**a, random forest and binary decision trees perform better than AdaBoost, and Multi-Layered Perceptron. Looking at the results aggregated for the time windows (25/50/100 and after/before/full), no relevant difference seems to exists by varying the time frame's width (**Fig. 3**b), while "after" window seems determining better results (**Fig. 3**c).



a b c

Fig. 3. Average accuracies of the four ML algorithms (a), the average accuracy of all the algorithms aggregated by time windows length (b) and time windows position (c).

A more focused analysis was performed by inspecting the performance of each ML algorithm according to the seven emotions. As reported in **Fig. 4**, it is evident that, in general, emotions like sadness, anger, fear, disgust and surprise are predicted with higher accuracy (in all cases, outperforming a random classifier that should have an accuracy of 33%); joy is instead predicted with a medium accuracy (around 40% in all cases), while contempt has lower accuracy in all cases (around 34%). At this link https://bit.ly/3yUtrcF, it is available an Excel file reporting the details of the performance of all the models, as well as a set of visualization we built for our analysis.

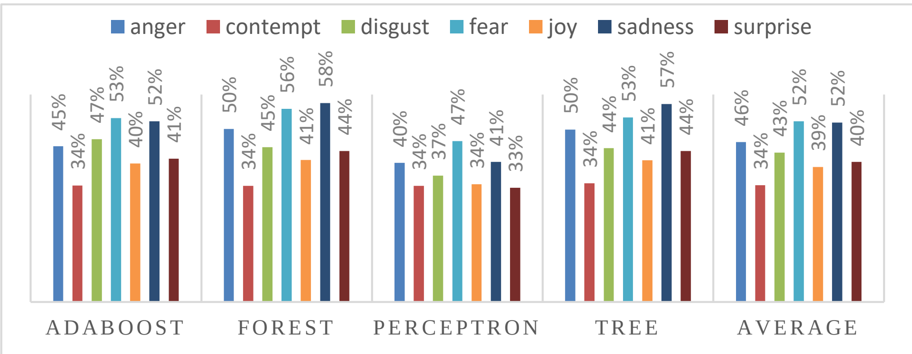


Fig. 4. Average accuracies of the four ML algorithms detailed for each emotion.

The results obtained at the end of the entire ML algorithm training process are promising. Regarding the emotion's prediction, it emerged that two algorithms, namely random forest and binary decision trees, on average, outperform AdaBoost, and Multi-Layered Perceptron. The limitations of the predictive models both in the accuracy levels can be explained by remembering that only a very small and random portion of the dataset (0.1%) was actually used.

Another interesting result comes from the time windows. For some algorithms, wider windows seem more informative and useful in creating more accurate models. The position of the window (before, after, before and after) does not seem the same. It would be interesting to compute more time windows in a wider time span to assess their effect on model accuracy, even considering the entire dataset extended with further data.

It seems that various algorithms perform better in different emotions (e.g.: random forest outperforms AdaBoost in predicting fear, but the contrary is true for disgust).

5 Conclusion and Future Works

The research work presented in this paper is about a full-automatic evaluation of UX. A tool that, starting from the user interaction logs of websites, detects users' emotions by exploiting ML algorithms is illustrated. Even if it is in an embryonic stage, it has great potential: the predictive models described can be easily embedded in a script to be provided to web developers so that, once inserted in a web page, collects users' interactions with the page and infers their emotions. This permits a page-by-page and point-by-point reconstruction of users' emotions, for example by using heatmaps.

As future work, we are going to complete the development of the software tool for the automatic detection of emotions. The average accuracy of ML algorithms like random forest and binary decision trees seem adequate and promising for this goal. However, a more accurate predictive model would be desirable: this could be obtained by either increasing the number of samples used in the training phase (the limitation to the 0.1% was only introduced to avoid a too long execution time) and by increasing the number of participants in the data-collection phase (thus expanding it "horizontally"). At the end of this research project, this dataset will be shared to foster similar studies. In addition, the spectrum of algorithms that have been compared may be broadened by including other classification algorithms like Support Vector Machines and k-Nearest Neighbors. Finally, models will be trained on emotions stratified in 5 and 7 classes, as proposed by SAM questionnaire. This could allow us a more detailed prediction. However, more classes will lower the prediction accuracy, but this will help us to establish a fair compromise between accuracy and detail of the prediction.

Acknowledgment

This work is partially supported by the Italian Ministry of University and Research (MIUR) under grant PRIN 2017 "EMPATHY: EMpowering People in deAling with internet of THings ecosYstems.".

References

1. Alibasa, M.J. and Calvo, R.A. (2019). Supporting Mood Introspection from Digital Footprints. In *Proc. of the International Conference on Affective Computing and Intelligent Interaction (ACII '19)*. IEEE.
2. Ardito, C., Buono, P., Caivano, D., Costabile, M.F., Lanzilotti, R., Bruun, A. and Stage, J. (2011). Usability evaluation: a survey of software development organizations. In *Proc. of the International Conference on Software Engineering*

and Knowledge Engineering (SEKE '11). Knowledge Systems Institute, Skokie, Illinois, USA., 282-287.

3. Atterer, R., Wnuk, M. and Schmidt, A. (2006). Knowing the user's every move. In *Proc. of the International Conference on World Wide Web (WWW '06)*. ACM.
4. Buono, P., Caivano, D., Costabile, M.F., Desolda, G. and Lanzilotti, R. (2020). Towards the Detection of UX Smells: The Support of Visualizations. *IEEE Access*, 8, 6901-6914.
5. Constantine, L., Badaro, G., Hajj, H., El-Hajj, W., Nachman, L., Bensaleh, M. and Obeid, A. (2016). A Framework for Emotion Recognition from Human Computer Interaction in Natural Setting. In *Proc. of the Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM.
6. Do, A.M., Rupert, A.V. and Wolford, G. (2008). Evaluations of pleasurable experiences: The peak-end rule. *Psychonomic Bulletin & Review*, 15(1), 96-98.
7. Edmonds, A. (2003). Uzilla: A new tool for Web usability testing. *Behavior Research Methods, Instruments, & Computers*, 35(2), 194-201.
8. Ekman, R. 1997. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA.
9. Federici, S., Mele, M.L., Lanzilotti, R., Desolda, G., Bracalenti, M., Buttafuoco, A., Gaudino, G., Cocco, A., Amendola, M. and Simonetti, E. (2019). Heuristic Evaluation of eGLU-Box: A Semi-automatic Usability Evaluation Tool for Public Administrations. In *Proc. of HCI International Conference (HCII '19)*. Springer International Publishing, 75-86.
10. Hartson, H.R., Castillo, J.C., Kelso, J. and Neale, W.C. (1996). Remote evaluation: the network as an extension of the usability laboratory. In *Proc. of the International Conference on Human Factors in Computing Systems (CHI '96)*. ACM, 228–235.
11. Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent data analysis*, 6, 429-449.
12. Khan, I.A., Brinkman, W.-P. and Hierons, R.M. (2008). Towards a Computer Interaction-Based Mood Measurement Instrument. In *Proc. of the Annual Workshop of the Psychology of Programming Interest Group (PPIG '08)*. ACM.
13. Kulke, L., Feyerabend, D. and Schacht, A. (2020). A Comparison of the Affectiva iMotions Facial Expression Analysis Software With EMG for Identifying Facial Expressions of Emotion. *Frontiers in Psychology*, 11(329).
14. Lanzilotti, R., Ardito, C., Costabile, M.F. and Angeli, A.D. (2011). Do patterns help novice evaluators? A comparative study. *Int. J. Hum.-Comput. Stud.*, 69(1–2), 52–69.
15. Law, E.L.-C., Roto, V., Hassenzahl, M., Vermeeren, A.P.O.S. and Kort, J. (2009). Understanding, scoping and defining user experience: a survey approach. In *Proc. of the Conference on Human Factors in Computing Systems (CHI '09)*. ACM, 719–728.
16. Loop11. *Online User Testing Made Easy | Loop11*. Retrieved from https://www.loop11.com/ Last Access April 07, 2021,
17. McDuff, D., Mahmoud, A., Mavadati, M., Amr, M., Turcot, J. and Kaliouby, R.e. (2016). AFFDEX SDK: A Cross-Platform Real-Time Multi-Face Expression Recognition Toolkit. In *Proc. of the Conference on Human Factors in Computing Systems - Extended Abstracts (CHI '16)*. ACM, 3723–3726.

18. Rubin, J. and Chisnell, D. 2008. *Handbook of usability testing: how to plan, design and conduct effective tests*. John Wiley & Sons.
19. Russo, P., Lanzilotti, R., Costabile, M.F. and Pettit, C.J. (2018). Towards satisfying practitioners in using Planning Support Systems. *Computers, Environment and Urban Systems*, 67, 9-20.
20. Salmeron-Majadas, S., Santos, O.C. and Boticario, J.G. (2014). An Evaluation of Mouse and Keyboard Interaction Indicators towards Non-intrusive and Low Cost Affective Modeling in an Educational Context. *Procedia Computer Science*, 35, 691-700.
21. Scholtz, J. (2004). Usability evaluation. *National Institute of Standards and Technology*, 1.
22. Shneiderman, B. 1980. *Software psychology: Human factors in computer and information systems (Winthrop computer systems series)*. Winthrop Publishers.
23. Sonderegger, A. and Sauer, J. (2009). The influence of laboratory set-up in usability tests: effects on user performance, subjective ratings and physiological measures. *Ergonomics*, 52(11), 1350-1361.
24. Stöckli, S., Schulte-Mecklenbeck, M., Borer, S. and Samson, A.C. (2018). Facial expression analysis with AFFDEX and FACET: A validation study. *Behavior Research Methods*, 50(4), 1446-1460.
25. Taggart, R.W., Dressler, M., Kumar, P., Khan, S. and Coppola, J.F. (2016). Determining emotions via facial expression analysis software. In *Proc. of the Student-Faculty Research Day (CSIS)*. Pace University.
26. Thompson, K.E., Rozanski, E.P. and Haake, A.R. (2004). Here, there, anywhere: remote usability testing that works. In *Proc. of the Conference on Information Technology Education (ITE '04)*. ACM, 132–137.
27. Userlytics. *Remote User Testing Platform | Userlytics*. Retrieved from https://www.userlytics.com/ Last Access April 07, 2021,
28. UserTesting. *UserTesting: The Human Insight Platform*. Retrieved from https://www.usertesting.com/ Last Access April 07, 2021,
29. Vea, L. and Rodrigo, M.M. (2017). Modeling Negative Affect Detector of Novice Programming Students Using Keyboard Dynamics and Mouse Behavior. *Lecture Notes in Computer Science.* Springer International Publishing, 127-138.
30. Zahidi, Z., Lim, Y.P. and Woods, P.C. (2014). Understanding the user experience (UX) factors that influence user satisfaction in digital culture heritage online collections for non-expert users. In *Proc. of the Science and Information Conference (SAI '14)*. IEEE, 57-63.
31. Zaina, L.A.M., Sharp, H. and Barroca, L. (2021). UX information in the daily work of an agile team: A distributed cognition analysis. *International Journal of Human-Computer Studies*, 147, 102574.
32. Zimmermann, P., Guttormsen, S., Danuser, B. and Gomez, P. (2003). Affective Computing—A Rationale for Measuring Mood With Mouse and Keyboard. *International Journal of Occupational Safety and Ergonomics*, 9(4), 539-551.