Visualizing Prediction Correctness of Eye Tracking Classifiers

Martin H.U. Prinzler
martin.prinzler@cs.uni-bremen.de
Database Lab,
University of Bremen
Bremen, Germany

Christoph Schröder
schroeder.c@cs.uni-bremen.de
Institute for Computer Graphics
and Virtual Reality,
University of Bremen
Bremen, Germany

Sahar Mahdie Klim Al Zaidawi
saharmah@cs.uni-bremen.de
Database Lab,
University of Bremen
Bremen, Germany

Gabriel Zachmann
zach@cs.uni-bremen.de
Institute for Computer Graphics
and Virtual Reality,
University of Bremen
Bremen, Germany

Sebastian Maneth
maneth@cs.uni-bremen.de
Database Lab,
University of Bremen
Bremen, Germany

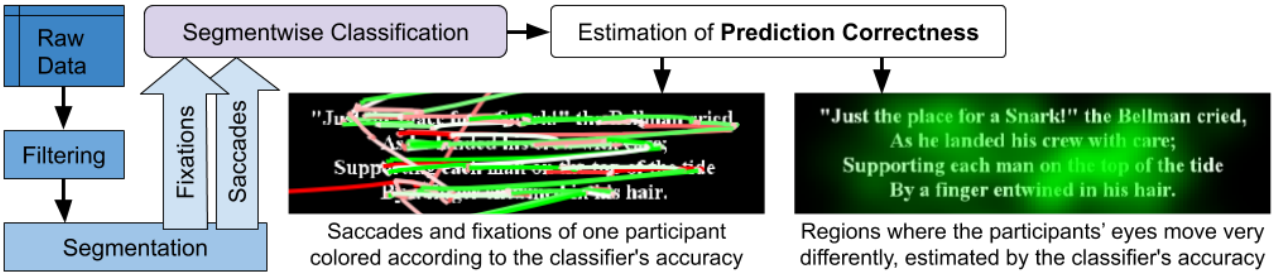


Figure 1: Flow chart from raw eye tracking data to two applications of the *Prediction Correctness Value*: trajectory based for a single participant (left) and heatmap based for multiple participants (right).

ABSTRACT

Eye tracking data is often used to train machine learning algorithms for classification tasks. The main indicator of performance for such classifiers is typically their prediction accuracy. However, this number does not reveal any information about the specific intrinsic workings of the classifier. In this paper we introduce novel visualization methods which are able to provide such information. We introduce the *Prediction Correctness Value* (PCV). It is the difference between the calculated probability for the correct class and the maximum calculated probability for any other class. Based on the PCV we present two visualizations: (1) coloring segments of eye tracking trajectories according to their PCV, thus indicating how beneficial certain parts are towards correct classification, and (2) overlaying similar information for all participants to produce a heatmap that indicates at which places fixations are particularly beneficial towards correct classification. Using these new visualizations we compare the performance of two classifiers (RF and RBFN).

ETRA '21 Short Papers, May 25–27, 2021, Virtual Event, Germany

ACM ISBN 978-1-4503-8345-5/21/05…$15.00
https://doi.org/10.1145/3448018.3457997

CCS CONCEPTS

• **Human-centered computing** → **Visualization**; *Heat maps.*

KEYWORDS

Eye Tracking, Explainable Artificial Intelligence; Prediction Visualization; Eye Movement Biometrics; Gaze Point Visualization; Machine Learning; User Identification;

ACM Reference Format:
Martin H.U. Prinzler, Christoph Schröder, Sahar Mahdie Klim Al Zaidawi, Gabriel Zachmann, and Sebastian Maneth. 2021. Visualizing Prediction Correctness of Eye Tracking Classifiers. In *ETRA '21: 2021 Symposium on Eye Tracking Research and Applications (ETRA '21 Short Papers), May 25–27, 2021, Virtual Event, Germany.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3448018.3457997

1 INTRODUCTION

Eye tracking data is often used for classification tasks; for instance, to determine users based on their eye movements [George and Routray 2016; Kasprowski and Ober 2004; Rigas and Komogortsev 2017; Schröder et al. 2020], to predict gender via eye movements

[Moss et al. 2012; Sargezeh et al. 2019; Zaidawi et al. 2020], or to predict some disorders such as dyslexia [Benfatto et al. 2016] (see also [Shojaeizadeh et al. 2019]). A central element is the used stimulus which shapes the behavior of participants of eye tracking experiments. This leads to two questions: (1) How do stimuli have to be designed to improve the performance of classification? and (2) What are the differences in the behavior of participants? Both questions are related to areas of the stimuli where the participants act most differently, and therefore the classifier works especially well. Understanding the classifier is related to the field of explainable artificial intelligence [Adadi and Berrada 2018; Arrieta et al. 2020; Samek et al. 2017]. Here we focus, instead of a concrete explanation of the algorithm, on a justification for its behavior (see [Biran and Cotton 2017] and [Nguyen et al. 2019]). To explore this topic a visualization of the classifier's performance could be helpful. While typically this means an overall assessment of the classifier (e.g., [Alaiz-Rodríguez et al. 2008; Seliya et al. 2009]) in our case the interest lies on single predictions with specific locations on the stimuli. There are already many different approaches to extract and visualize information regarding eye tracking data, see, e.g., [Blascheck et al. 2014, 2017] for an overview. Usually each manufacturer of an eye tracking device provides their own software (e.g., Tobii Pro AB [2014], GAZE INTELLIGENCE [2020], or S.R. Research Ltd. [2020]). There are commercial tools for multiple eye trackers (e.g., *GazeTracker* [Eyetellect 2016]) and there are open source tools (e.g., *Pupil* [Kassner et al. 2014], *OGAMA* [Voßkühler et al. 2008], *PyGaze Analyzer* [Dalmaijer et al. 2014], *EyeVis* [Menges et al. 2020], or *IRIS* [D'Angelo et al. 2019]). Recently also web tools arise [Bakardzhiev et al. 2020]. All of these tools provide a visualization of *fixations* and *saccades*. Most of them can present heatmaps, and have other specialties. To the best of our knowledge, none of the existing tools provides visualizations of the results of classification algorithms for eye tracking data.

In this paper we introduce a novel measure: the *Prediction Correctness Value* (PCV). It can be used to spatially visualize the correctness of predictions made by a classifier which in turn helps the user to understand the workings of the classifier better. We also find that these visualizations spawn new hypotheses which were not apparent to us without the visualizations. Given sample data and a set of classes, a classification algorithm computes a probability distribution over the classes. The PCV is defined as the calculated probability for the correct class minus the maximum calculated probability for any other class. The PCV tells us if and how well a classifier was able to make a prediction (when the PCV is negative, the prediction was wrong). In this paper we present two techniques for visualizing the PCV: (1) trajectory based visualization for single participants, and (2) heatmap based visualization for several participants. The *Prediction Correctness Trajectory* (PCT) lets us focus on one participant at a time. It shows in detail which eye movements of a participant caused the classifier to make a correct decision. To have a better overview of a complete dataset we present the *Prediction Correctness Heatmap* (PCH). It combines the predictions of several participants in relation to the used stimulus showing which regions are beneficial for correct classification and which are not. E.g., for a reading stimulus, the heatmap can highlight single words or syllables where the participants act very differently (with respect to the classes) and therefore can be classified well (see Figure 1), which leads to high prediction correctness.

To present our visualization methods we focus on one concrete classification task (biometrics). The classifier is supposed to identify people via their eye movements. As is commonly done (see, e.g., [George and Routray 2016]) we feed the classifiers with feature vectors of trajectory segments that represent fixations and saccades. We focus on two classifiers in this paper: *Random Decision Forests* (RF) [Breiman 2001] and *Radial Basis Function Networks* (RBFN) [Broomhead and Lowe 1988].

The paper is organized as follows: Description of the used Dataset is in Section 2 followed by the description of the methods (namely Filtering, Segmentation, Classification, and Features) in Section 3. In Section 4 the new visualization techniques are explained. Section 5 shows some applications. The paper closes with conclusions and future work in Section 6.

2 DATASETS

We use two datasets from the 2015 *BioEye competition* [Rigas and Komogortsev 2017]. Both contain data obtained from 153 participants, whose tasks were to read a poem (TEX), and to observe a randomly moving dot (RAN). For the TEX stimulus, there are two 60 seconds recordings per participant which were recorded with a pause of 30 minutes in between. For the RAN stimulus, there are also two recordings, each of length 100 seconds. All sessions were recorded with an EyeLink-1000 eye-tracker at 1000 Hz and were decimated to 250 Hz to have a balance between noise filtering, data size and preservation of the eye movement characteristics (see [Rigas and Komogortsev 2017] for further details). The participants are comprised of males and females aged 18 to 46. During the recordings, the head of each participant was positioned on a chin rest at a distance of 550 mm from a 22-inch screen (resolution 1680 x 1050).

In the TEX dataset the alignment of the gaze trajectories to the stimulus is not correct. This is obvious because of the specific spacing of the text (primarily by the distances between heading, paragraphs and lines). We performed horizontal and vertical corrections for each user in each session to fit the trajectory plausible to the poem. This was done by hand and is therefore subjective but certainly improves the alignment. The procedure sharpens all results which depend on the position of the trajectory.

3 METHODS

In this section, we describe the steps from the raw data to the predictions of our classifiers (see also Figure 1).

Filtering. To reduce noise and other undesirable artifacts we apply a Savitzky-Golay filter [Savitzky and Golay 1964] (see also [Schafer 2011]). This filter is controlled by two parameters: the number N of the considered samples (*filter width*) and the *degree* D of the used polynomial. For every point (x_i, y_i), with $(N-1)/2 \leq i \leq n-(N-1)/2$, where n is the total number of points, the filter fits a symmetric polynomial of degree D through the amount of selected samples N. The point (x_i, y_i) is always the center of the selected samples, so only odd filter widths N are allowed.

We use $N = 7$ and $D = 1$, which reduces the noise without influencing the trajectory too much. With this setting the filter works like a line fit starting 3 points before and ending 3 points after the point to calculate.

Segmentation. To divide the gaze trajectories into fixations and saccades, we implemented the simple *Identification-by-Velocity-Threshold* (IVT) algorithm which is described in slightly different ways in multiple publications (e.g., [Erkelens and Vogels 1995; Sen and Megaw 1984], and [George and Routray 2016]). Our implementation of the IVT algorithm uses two parameters (like [George and Routray 2016]): the *velocity threshold* (VT) and the *minimal fixation time threshold* (FT). The algorithm defines as fixation all consecutive gaze points resulting in eye rotation velocities below the VT, unless the fixation would be shorter than the FT. All other segments are identified as saccades. We use VT = 15 deg/s and FT = 50 ms. The values were chosen by hand so that plausible numbers of fixations appear.

Classification. In our context, classification means to label eye tracking data with the ID of a unique participant. If L is the set of participant IDs, then the classification task is to learn and predict a function p from trajectories t to probability distributions over L: $p(t) : L \rightarrow (0, 1)$. For a given ID $u \in L$, $p(t, u)$ denotes the probability for participant u. For us, p is determined by learning two such functions p_f and p_s (one for fixations and one for saccades) and averaging them to create a single result. As classifiers we use *Random Decision Forests* (RF) (as implemented in [Pedregosa et al. 2011]) with 200 estimators and an implementation of *Radial Basis Function Networks* (RBFN) with 32 clusters as described in [George and Routray 2016]. In both datasets we have two sessions for each user: one is used for training and the other for testing. Unless otherwise stated, all results we presente are from the test session.

Features. We calculate a set of 9 fixation and 43 saccade features as described by George and Routray [2016] from the fixation and saccade segments to feed into the classification algorithms. These include, for example, features like duration, path length, angular velocity, and statistical features such as standard deviation, skewness, or kurtosis, but also features related to the previous or next segment like distance or angle.

4 VISUALIZING THE CORRECTNESS OF PREDICTIONS

In this section we describe how we calculate the correctness of a prediction and how we visualize it. All results presented in this paper were created with our own tool written in Python. It includes an interactive GUI to visualize and prepare eye tracking data using the Bokeh library [Bokeh Development Team 2018] and is available as open source from the url: http://wwwdb.informatik.uni-bremen.de/smida_pcv/.

4.1 Calculation of the Prediction Correctness

For a given trajectory segment t, our classifier returns a probability $p(t, u)$ for each participant u. Let $c(t)$ be the participant who produced the segment t, i.e.; it is the correct class that the classifier should choose. We introduce the *Prediction Correctness Value* (PCV),

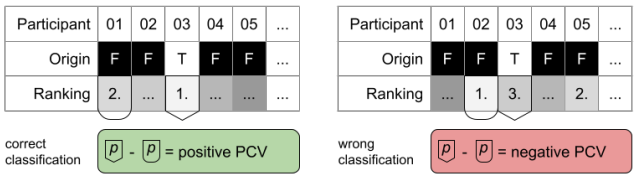


Figure 2: Explanation of the *Prediction Correctness Value* (PCV) at a segment for an excerpt of five participants. The segment belongs to participant *03* and is classified correctly on the left side. The algorithm on the right side predicted participant *02* as the segment's creator. The calculated probability of the correct participant is subtracted either by the second ranked guess of the classifier, or in case of a wrong prediction, by the probability for the first ranked guess.

which is the difference between the calculated probability of the participant $c(t)$, and the highest probability from any other participant $p_m(t) = \max \{p(t, u) \mid u \in L \backslash \{c(t)\}\}$:

$$\mathrm{PCV}(t) = p(t, c(t)) - p_m(t).$$

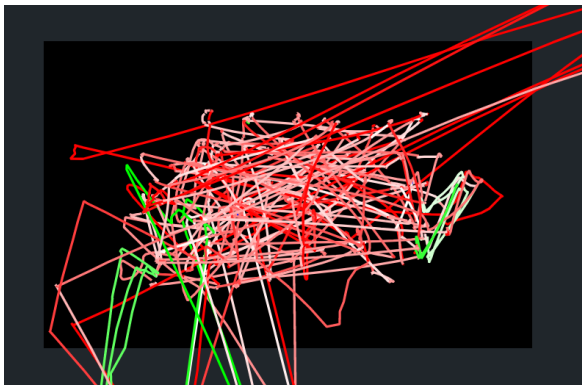
The concept is visualized in Figure 2. In case of a correct prediction, the PCV is positive. If the classifier predicted any other participant, the PCV will be negative. The greater the difference from the first to the second guess of the algorithm, the greater the absolute value of the PCV. So high absolute values mean high confidence of the classifier in its decision.

4.2 Prediction Correctness Trajectory for Single Participants

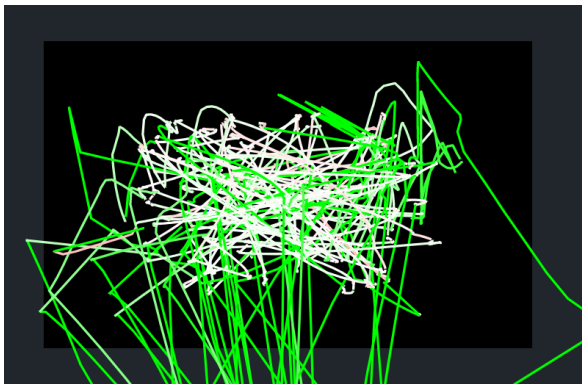
As a simple example, we consider the PCVs for single participants. Be aware that our calculations are based on one run of one classifier. The result will vary with different settings.

Figure 3 shows an example for a single participant (*ID_053*) from the RAN dataset. The continuous line is the actual gaze trajectory colored according to the PCV. We call this a *Prediction Correctness Trajectory*. Green means positive PCV, white means close to zero, and red negative. The full saturation of the color is reached for the top 10 % of the PCVs. Most of the main movements for the task (following the dots) are wrongly classified in the test case (a). For all paths outside of the stimulus region in the top right, the classifier predicts the wrong participants even with high confidence. Nevertheless, the used RF algorithm could use the lower left paths to identify the correct participant. In this case, in the training data (b), the participant had many outgoing paths at the bottom, but none at the top, which is an explanation of the classifier's behavior.

In Figure 4 the PCV is shown for a participant (*ID_045*) from the TEX dataset reading a poem. The upper two images (a, b) show the actual test case, where the algorithms have not seen the data before. The training happened on the data shown in the lower two images (c, d). On the left (a, c), the applied classifier is RF, while on the right (b, d) it is RBFN. It is visible that RF is overfitted and identifies every segment correctly in the training data. RBFN instead performs similarly on the training and on the test data. While the outliers are correctly identified in the training data, they



(a) Test case



(b) Training case

Figure 3: Visualization of correctness of predictions made by a *Radial Basis Function Networks* classifier on the RAN stimulus (the participant is following random appearing dots). The participant is correctly identified in the test case (a) by the outgoing paths to the bottom, which are also present in the training case (b).

are mistaken for a different participant by both classifiers in the testing cases. In contrast, eye movements in the region of the text are mostly correctly classified.

4.3 Prediction Correctness Heatmap for Multiple Participants

The consideration of results for single participants can bring up a detailed knowledge of the classifiers and anomalies in the data but needs investigation. Occurring patterns may depend not only on the single participant but also on the combination of participants used for training the algorithm. When we consider multiple participants at once we obtain insights on more general patterns. Minor differences in the starting conditions, which can affect the prediction of single images, will average to a more stable outcome when considering multiple images at once.

To do so, we acquire the centers of gravity of all fixations and calculate a two dimensional histogram, where we sum up the PCVs. We call this the *Prediction Correctness Heatmap* (PCH). For a bin i, j of the histogram and with a total number of n_{fix} fixations this means:

$$\mathrm{PCH}_{i,j} = \sum_{k=1}^{n_{fix}} \begin{cases} \mathrm{PCV}_k & \text{if fixation}_k \text{ in bin}_{i,j} \\ 0, & \text{otherwise.} \end{cases}$$

After the calculation of 500×500 bins we use a Gaussian filter ($\sigma = 5$, implemented by SciPy [Virtanen et al. 2020]) to blur the image for a more natural look. We distinguish fixations with a positive PCV from these with a negative PCV.

In Figure 5 we show the positive histograms for the 153 participants of the TEX dataset, which are classified by RF (left: a, c) and RBFN (right: b, e). The top row (a, b) shows results from the test cases with unseen data. For the bottom row (c, e), the algorithms were applied to the data they were trained with. The values are visualized in green by a color scale from transparent (zero) to 90 % opacity (maximum).

It is clear that the overall occurrence follows a standard density heatmap of the fixations. This is shown in image (d) in Figure 5. The frequency of fixations is visualized in yellow by a color scale from transparent (none) to 90 % opacity (maximum). While the first paragraph is covered with 45 fixations on average and the second and the third have still around 40, from the fourth to the sixth paragraphs there are less fixations. The reason is that some participants do not finish the complete poem and others start over again. By comparing the fixation heatmap (d) with the PCH images, we find that in the bottom paragraph, there are less beneficial predictions because there are less fixations in total. Furthermore, the PCH on the training data (c, e) in Figure 5 looks similar to the general fixation heatmap. This is especially the case for the prediction of RF (c) because it is overfitting and predicting nearly every fixation correctly. RBFN, on the other hand, has a slightly different pattern. It seems to prefer fixations in some regions over others (see the more intense color in the first and last paragraph). Note: The maximum opacity is related to the different distribution of values for each image and can only be compared qualitatively.

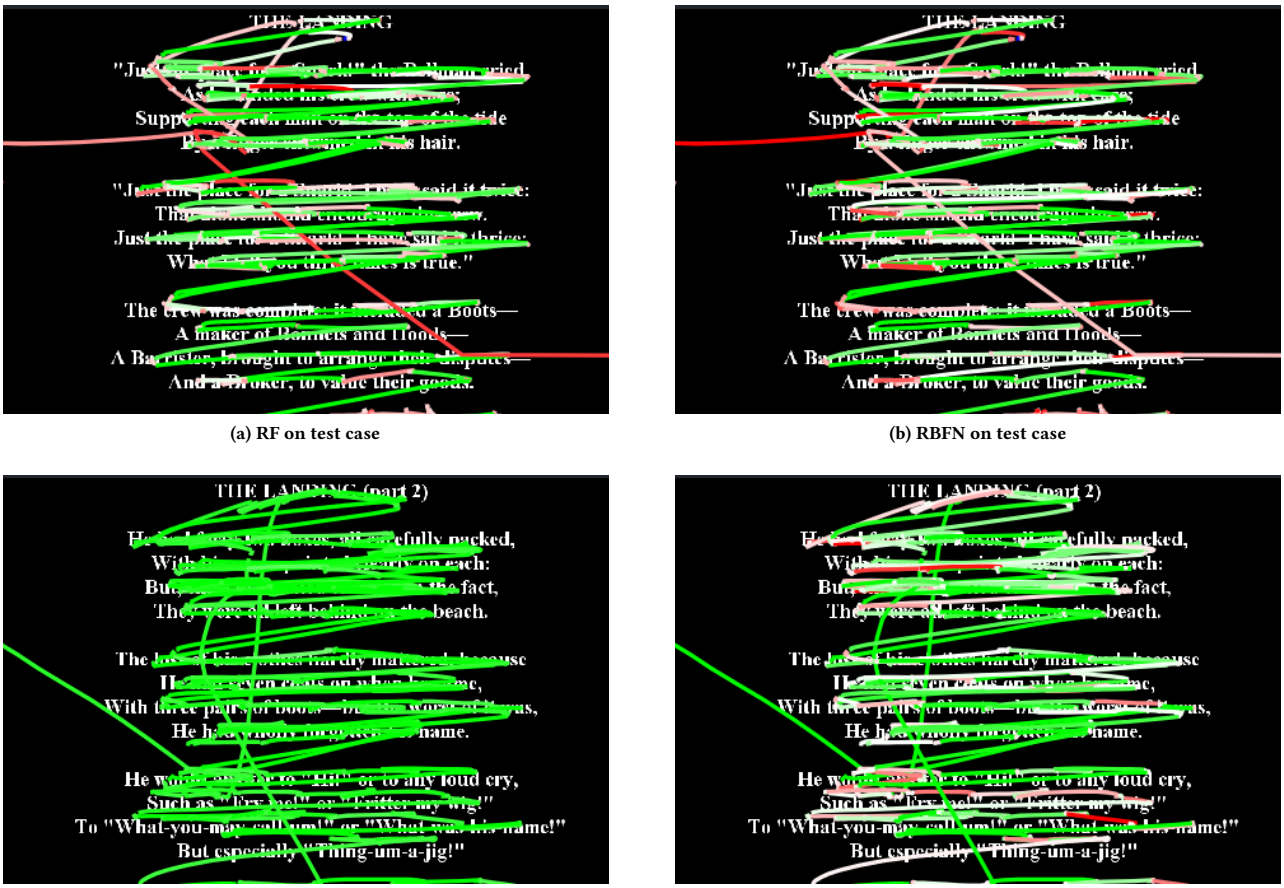
By viewing the test case (a, b) in Figure 5, we find there is a clear pattern for beneficial fixations, and it is not dependent on the classifier. However, the interpretation of the pattern is open to discussion (see Section 5).

Note: In our experience, saccades contribute more to the classification than fixations, but the calculation of heatmaps for saccades is more difficult since saccades cannot easily be consolidated to one point. Using all the samples of the saccades and applying our present method, we found no specific patterns. The PCHs calculated in this way is similar to the general saccade heatmap, showing only the saccade density.

5 APPLICATIONS

5.1 Prediction Correctness Trajectory

Let us demonstrate how PCTs can be used to generate hypotheses about the eye tracking data: In Figure 3(a) we observe that all paths that leave the stimulus window to the top-right are colored red. On the other hand, of those paths that leave the stimulus window to the bottom, some are colored green and others red. At first, we thought that these paths might belong to two different groups



(a) RF on test case

(b) RBFN on test case

(c) RF on training case

(d) RBFN on training case

Figure 4: Visualization of correctness of predictions made by a *Random Decision Forests* (left) and a *Radial Basis Function Networks* classifier (right) on the TEX stimulus. Both classifiers perform equally in the test case (top), while RF (left) overfits in the training case (bottom).

(e.g., two different kind of blinkings), but Figure 3(b) reveals that the explanation may be much simpler: the training data does not contain any paths that lead to the top-right. Consider Figure 4. We can see that in the training cases the two classifiers (RF and RBFN) behave *quite differently*: in the case of RF, all segments are colored green (which indicates overfitting), while in the case of RBFN several segments are colored white or red. Nevertheless, and we find this astonishing, the behavior of both classifiers on the test data is *rather similar*: segments that are white or red in Figure 4(a) (i.e., for RF) are also white or red in Figure 4(b). Similarly, segments that are green in Figure 4(b) are also green in Figure 4(a).

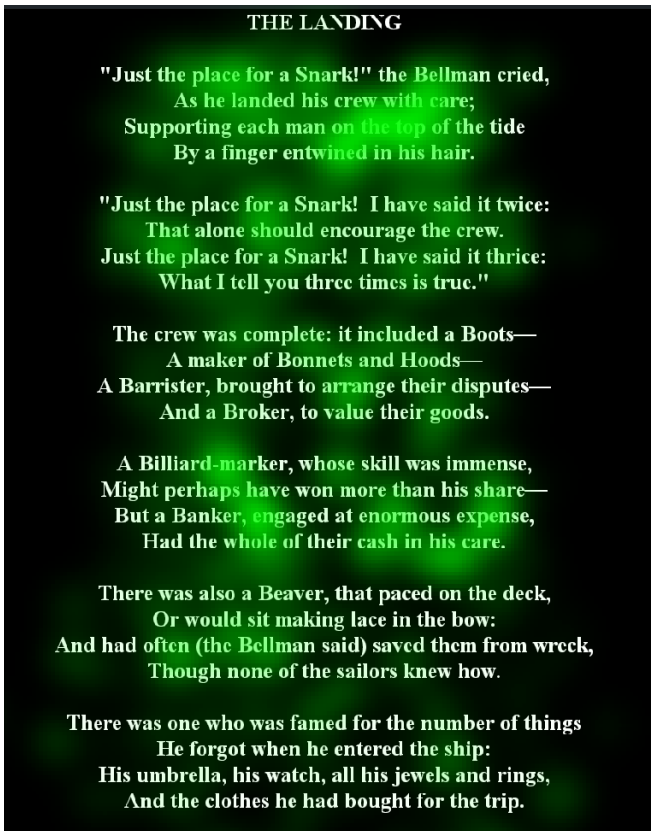
5.2 Prediction Correctness Heatmap

The PCH combines the PCV of multiple users into one image. Consider Figure 5 which combines the fixations of all participants. We see that RF and RBFN behave differently on the training case: for RBFN, more green regions are on the left or the bottom part of the poem, while for RF more green regions are towards the top and are evenly spread from left to right. Again it turns out that in the test case, both classifiers behave very similarly! And more than that, let us consider the regions that are colored rich green in the test cases of the two classifiers: it appears that these regions are on simpler and shorter words, rather than on more complex words. This could lead to the hypothesis that participants differ more reliably in their "common reading behavior" rather than in their approach to understanding complex words.

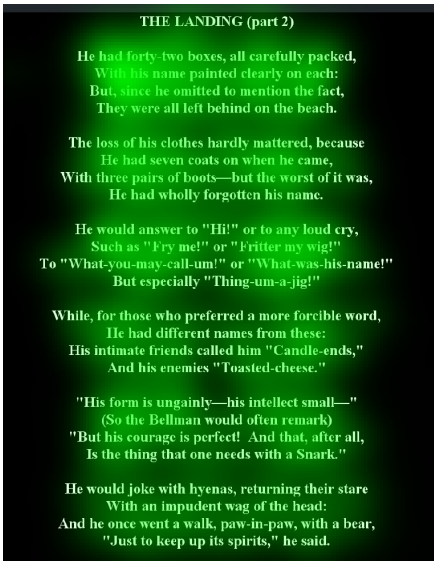
6 CONCLUSIONS AND FUTURE WORK

We consider classification tasks over eye tracking data. We define the *Prediction Correctness Value* (PCV) as the difference between the calculated probability for the actual correct class and the highest calculated probability for any other class. We then present two ways of visualizing PCVs: the *Prediction Correctness Trajectory* (PCT) in which segments are colored according to their PCV (we use green for positive PCVs and red for negative PCVs) and the *Prediction Correctness Heatmap* (PCH) which combines the PCTs of several

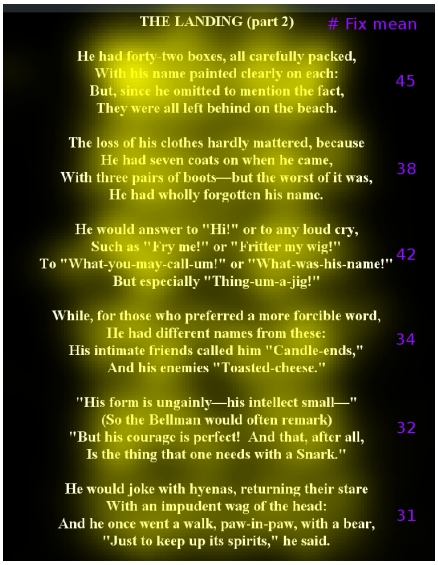
(a) PCH with RF on test case



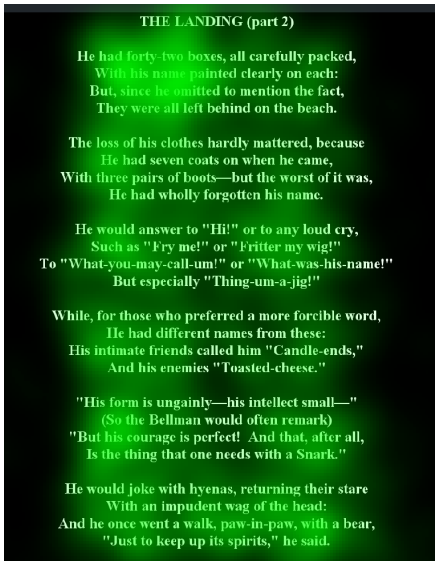
(b) PCH with RBFN on test case



(c) PCH with RF on training case



(d) fixations heatmap



(e) PCH with RBFN on training case

Figure 5: Beneficial fixation areas for predicting the correct participant via *Random Decision Forests* (left) and *Radial Basis Function Networks* classifier (right) in the TEX dataset. Figure (d) shows the overall distribution of fixations in yellow. The average number of fixations per user is written in purple beside each paragraph.

users. Both visualization methods give insights into the workings of the different classifiers. For instance, overfitting is easily observed by only green segments in the training data. The PCH of our poem reading stimulus shows particular regions and words that are beneficial for the classifiers. We expect that many more interesting observations can be made on other data sets using PCTs and PCHs. For instance, they may reveal particular groups of "outliers" (e.g., paths that exit the stimulus window) and how these influence the working of the classifier. In the future we would like to apply our methods to other classifiers and to datasets with other stimuli; moreover, we would like to investigate ways to produce heatmaps for saccades.

ACKNOWLEDGMENTS

We thank Oleg Komogortsev for providing the used dataset and the anonymous reviewers for their helpful comments.

REFERENCES

Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE access* 6 (2018), 52138–52160.

Rocio Alaiz-Rodríguez, Nathalie Japkowicz, and Peter Tischer. 2008. Visualizing classifier performance on different domains. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, Vol. 2. IEEE, 3–10.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115.

Hristo Bakardzhiev, Marloes Burgt, Eduardo Martins, Bart Dool, Chyara Jansen, David Scheppingen, Günter Wallner, and Michael Burch. 2020. A Web-Based Eye Tracking Data Visualization Tool.

Mattias Nilsson Benfatto, Gustaf "Oqvist Seimyr, Jan Ygge, Tony Pansell, Agneta Rydberg, and Christer Jacobson. 2016. Screening for dyslexia using eye tracking during reading. *PloS one* 11, 12 (2016), e0165508.

Or Biran and Courtenay Cotton. 2017. Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)*, Vol. 8. 8–13.

Tanja Blascheck, Kuno Kurzhals, Michael Raschke, Michael Burch, Daniel Weiskopf, and Thomas Ertl. 2014. State-of-the-Art of Visualization for Eye Tracking Data. In *Eurographics Conference on Visualization, EuroVis 2014 - State of the Art Reports, STARs, Swansea, UK, June 9-13, 2014*, Rita Borgo, Ross Maciejewski, and Ivan Viola (Eds.). Eurographics Association. https://doi.org/10.2312/eurovisstar.20141173

Tanja Blascheck, Kuno Kurzhals, Michael Raschke, Michael Burch, Daniel Weiskopf, and Thomas Ertl. 2017. Visualization of Eye Tracking Data: A Taxonomy and Survey. *Comput. Graph. Forum* 36, 8 (2017), 260–284. https://doi.org/10.1111/cgf.13079

Bokeh Development Team. 2018. *Bokeh: Python library for interactive visualization*. https://bokeh.pydata.org/en/latest/

Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32. https://doi.org/10.1023/A:1010933404324

David S. Broomhead and David Lowe. 1988. Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems* 2, 3 (1988).

Edwin S. Dalmaijer, Sebastiaan Mathôt, and Stefan Van der Stigchel. 2014. PyGaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments. *Behavior research methods* 46, 4 (2014), 913–921.

Sarah D'Angelo, Jeff Brewer, and Darren Gergle. 2019. Iris: a tool for designing contextually relevant gaze visualizations. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, ETRA 2019, Denver , CO, USA, June 25-28, 2019*, Krzysztof Krejtz and Bonita Sharif (Eds.). ACM, 79:1–79:5. https://doi.org/10.1145/3317958.3318228

Casper J Erkelens and Ingrid MLC Vogels. 1995. The initial direction and landing position of saccades. In *Studies in Visual Information Processing*. Vol. 6. Elsevier, 133–144.

Eyetellect. 2016. *GazeTracker*. http://www.eyetellect.com/gazetracker/

GAZE INTELLIGENCE. 2020. *Blickshift software*. https://gazeintelligence.com/blickshift-analytics-1

Anjith George and Aurobinda Routray. 2016. A score level fusion method for eye movement biometrics. *Pattern Recognition Letters* 82 (2016), 207–215. https://doi.org/10.1016/j.patrec.2015.11.020

Pawel Kasprowski and Józef Ober. 2004. Eye Movements in Biometrics. In *Biometric Authentication, ECCV 2004 International Workshop, BioAW 2004, Prague, Czech Republic, May 15, 2004, Proceedings*. 248–258. https://doi.org/10.1007/978-3-540-25976-3_23

Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *The 2014 ACM Conference on Ubiquitous Computing, UbiComp '14 Adjunct, Seattle, WA, USA - September 13 - 17, 2014*, A. J. Brush, Adrian Friday, Julie A. Kientz, James Scott, and Junehwa Song (Eds.). ACM, 1151–1160. https://doi.org/10.1145/2638728.2641695

Raphael Menges, Sophia Kramer, Stefan Hill, Marius Nisslmueller, Chandan Kumar, and Steffen Staab. 2020. A visualization tool for eye tracking data analysis in the web. In *ACM Symposium on Eye Tracking Research and Applications*. 1–5.

Felix Joseph Mercer Moss, Roland Baddeley, and Nishan Canagarajah. 2012. Eye movements to natural images as a function of sex and personality. *PLoS One* 7, 11 (2012), e47870.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2019. Understanding neural networks via feature visualization: A survey. In *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer, 55–76.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.

Ioannis Rigas and Oleg V. Komogortsev. 2017. Current research in eye movement biometrics: An analysis based on BioEye 2015 competition. *Image Vision Comput.* 58 (2017), 129–141. https://doi.org/10.1016/j.imavis.2016.03.014

Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. 2017. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296* (2017).

Bahman Abdi Sargezeh, Niloofar Tavakoli, and Mohammad Reza Daliri. 2019. Gender-based eye movement differences in passive indoor picture viewing: An eye-tracking study. *Physiology & behavior* 206 (2019), 43–50.

Abraham. Savitzky and M. J. E. Golay. 1964. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry* 36, 8 (1964), 1627–1639. https://doi.org/10.1021/ac60214a047

Ronald W. Schafer. 2011. What Is a Savitzky-Golay Filter? [Lecture Notes]. *IEEE Signal Process. Mag.* 28, 4 (2011), 111–117. https://doi.org/10.1109/MSP.2011.941097

Christoph Schröder, Sahar Mahdie Klim Al Zaidawi, Martin H. U. Prinzler, Sebastian Maneth, and Gabriel Zachmann. 2020. Robustness of Eye Movement Biometrics Against Varying Stimuli and Varying Trajectory Length. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik (Eds.). ACM, 1–7. https://doi.org/10.1145/3313831.3376534

Naeem Seliya, Taghi M Khoshgoftaar, and Jason Van Hulse. 2009. A study on the relationships of classifier performance metrics. In *2009 21st IEEE international conference on tools with artificial intelligence*. IEEE, 59–66.

Tayyar Sen and Ted Megaw. 1984. The effects of task variables and prolonged performance on saccadic eye movement parameters. In *Advances in Psychology*. Vol. 22. Elsevier, 103–111.

Mina Shojaeizadeh, Soussan Djamasbi, Randy C. Paffenroth, and Andrew C. Trapp. 2019. Detecting task demand via an eye tracking machine learning system. *Decis. Support Syst.* 116 (2019), 91–101. https://doi.org/10.1016/j.dss.2018.10.012

S.R. Research Ltd. 2020. *Data Viewer*. https://www.sr-research.com/data-viewer/

Tobii Pro AB. 2014. *Tobii Pro Lab*. Danderyd, Stockholm. http://www.tobiipro.com/

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Adrian Voßkühler, Volkhard Nordmeier, Lars Kuchinke, and Arthur M Jacobs. 2008. OGAMA (Open Gaze and Mouse Analyzer): open-source software designed to analyze eye and mouse movements in slideshow study designs. *Behavior research methods* 40, 4 (2008), 1150–1162.

Sahar Mahdie Klim Al Zaidawi, Martin H. U. Prinzler, Christoph Schröder, Gabriel Zachmann, and Sebastian Maneth. 2020. Gender Classification of Prepubescent Children via Eye Movements with Reading Stimuli. In *Companion Publication of the 2020 International Conference on Multimodal Interaction, ICMI Companion 2020, Virtual Event, The Netherlands, October, 2020*, Khiet P. Truong, Dirk Heylen, Mary Czerwinski, Nadia Berthouze, Mohamed Chetouani, and Mikio Nakano (Eds.). ACM, 1–6. https://doi.org/10.1145/3395035.3425261