

The Gaze and Mouse Signal as Additional Source for User Fingerprints in Browser Applications

Wolfgang Fuhl^{1,*}, Daniel Weber¹ and Shahram Eivazi^{1,2}

¹University Tübingen, Sand 14, Tübingen, Germany

²FESTO, Rüter Str. 82, Esslingen am Neckar, Germany

Keywords: User Identification, Gaze vs Mouse Movements, Studie, Machine Learning, Classification, Browser Fingerprint.

Abstract: In this work, we inspect different data sources for browser fingerprints. We show which disadvantages and limitations browser statistics have and how this can be avoided with other data sources. Since human visual behavior is a rich source of information and also contains person specific information, it is a valuable source for browser fingerprints. However, human gaze acquisition in the browser also has disadvantages, such as inaccuracies via webcam and the restriction that the user must first allow access to the camera. However, it is also known that the mouse movements and the human gaze correlate and therefore, the mouse movements can be used instead of the gaze signal. In our evaluation, we show the influence of all possible combinations of the three information sources for user recognition and describe our simple approach in detail.

1 INTRODUCTION

User identification plays a crucial role in many industrial sectors. In its original form, it is used to protect data and access to networks or premises (Lee et al., 2010; Choubey and Choubey, 2013). Today, there is a growing need for user identification, especially in the online environment, which includes both personalized advertising (Tucker, 2014) and product placement (Shamdasani et al., 2001; Fossen and Schweidel, 2019), but also online banking (Lee et al., 2010) or external access to corporate networks (Cole, 2011). For security-critical applications such as external access to company networks or online banking, user IDs and passwords have become widely accepted. When using security-critical functionalities, additional security prompts such as a generated PIN or SMS prompts are added. In online advertising, as well as product placement, companies try to identify a person without accessing critical personal data. This is guaranteed in the modern world by so-called cookies (Juels et al., 2006) since those have to be activated by the user, stateless approaches only use browser statistics (Juels et al., 2006). A disadvantage of this method is that the statistics can be used to identify a computer very effectively, but in the case of computers with multiple

users, all of them are treated as the same person. For the password and user recognition procedure, there are also disadvantages. For example, if the identification and password is known by an attacker, the attacker can gain access.

In this work, we analyze new data sources, like the eye signal and mouse movements. The basic idea is that a person can be identified by means of gaze signals or human visual behavior. This has been shown several times (Holland and Komogortsev, 2011; Fuhl et al., 2019; Fuhl et al., 2020). Since the gaze signal can only be computed with a webcam in a browser, it requires the user to activate and allow the access to the webcam. Additionally, the quality of the camera as well as the different lighting conditions influence the accuracy of the gaze signal (Papoutsaki et al., 2016). Further scientific work has already been done on the correlation of mouse movements and the human eye signal (Liebling and Dumais, 2014; Guo and Agichtein, 2010; Navalpakkam et al., 2013). It has been found that when clicking or the end point of a mouse movement almost always corresponds to the eye position (Liebling and Dumais, 2014; Guo and Agichtein, 2010; Navalpakkam et al., 2013). With this information, the technique of webcam based eye tracking has changed in that the mouse information is used to calibrate the eye tracker (Papoutsaki et al., 2016). Another advantage of mouse

*Corresponding author

movements is that this information is freely accessible in the browser and does not have to be activated manually by the user like the camera. We show in this thesis that the mouse information is sufficient to identify a user, which is also scientifically based on the fact that visual behavior is user-specific (Holland and Komogortsev, 2011; Fuhl et al., 2019; Fuhl et al., 2020) and that mouse movements in the browser correlate with visual behavior (Liebling and Dumais, 2014; Guo and Agichtein, 2010; Navalpakkam et al., 2013).

The application of these data sources in the industrial environment is enormous. For example, it enables continuous user validation for online banking and external access to corporate networks. It would not be enough to have only the password and the user ID, one would also have to be able to emulate the behavior of the correct human user. For user-specific advertising and product placement, it is also possible to differentiate between users on a shared computer and identification of the same user on different computers.

2 RELATED WORK

In this section, we discuss the state of the art regarding browser-based user identification. The first work which has dealt with browser-based user identification is (Mayer, 2009). It analyzed and used statistics about the browser configuration, version and installed extensions. In (Eckersley, 2010) the approach was proven in a larger study, and thus it was shown that the digital fingerprint can be effectively used for user identification via statistics. Further, studies (Alaca and Van Oorschot, 2016; Englehardt and Narayanan, 2016; Laperdrix et al., 2020; Kobusińska et al., 2018) dealt with extensions of the statistical features and their quality for user recognition. In (Alaca and Van Oorschot, 2016; Kobusińska et al., 2018) an analysis of the stability of the individual characteristics was also carried out. (Laperdrix et al., 2020; Kobusińska et al., 2018) examined different browsers and also analyzed security settings that can prevent fingerprinting. There was also a long term study which dealt with the creation of a unique fingerprint over years (Gómez-Boix et al., 2018). Cross-browser fingerprinting was covered in (Cao et al., 2017) using both operating system and hardware features. A further extension of these approaches is the use of hashing algorithms to make the calculation and identification more effective (Gabryel et al., 2020).

Applications for browser-based fingerprints are described in the literature as user tracking (Eckers-

ley, 2010; Englehardt and Narayanan, 2016), abuse prevention (Vastel et al., 2020), and authentication (Alaca and Van Oorschot, 2016) in many contexts. For example, security companies use the fingerprint to detect bots or abnormal behavior on web pages (mis, 2020b; mis, 2020a). In (Vastel et al., 2020) it is also shown that fingerprinting can be used to easily block scripts that collect data from web pages, but the authors also show that this protection can be easily circumvented. A fingerprint for mobile devices, which was calculated on all hardware and installed software, is described in (Bursztein et al., 2016). This makes it possible to distinguish between the real device and a simulated environment of the same device, and thus block network traffic in case of a simulated device.

Literature that deals with abuse prevention is mostly in the context of advertising or e-commerce. These concerns click fraud or credit card payments. Two new inference techniques were presented in (Nagaraja and Shah, 2019). The first technique recognizes click patterns within an advertising network and thus can prevent click fraud. In the second technique, bait clicks are injected and resulting conspicuous patterns are detected. (Renjith, 2018) deals with credit card fraud. Here, cheap goods or services are sought that have never been shipped or performed. The authors use different features and machine learning algorithms to detect this type of fraud.

There is also already some work in the field of deep neural networks for fraud detection. In (Zhang et al., 2019) a deep neural network was presented, which analyses the data for similar behavior. This allows fraud cases, which are repeated and follow the same procedures, to be detected and traced. This technique also helps to protect against known fraud, because the behavior is conspicuous for the system. Several interconnected neural networks have also been used to detect intrusion into computer networks (Ludwig, 2019). Here, various deep neural networks are used to monitor network communication. These networks detect patterns in the communication which do not correspond to the norm and warn early in case of a possible intrusion. Deep Boltzmann machines were used for fraud detection in biometric systems (de Souza et al., 2019). For this purpose, features from the deep layers of the network were used, as these have proven to be more robust against attempts of fraud. Another use of deep neural networks for fingerprint calculation is described in (Salakhutdinov and Hinton, 2009). Here, auto-encoders are used to calculate a hash of a document. Similar documents produce a similar hash. This technique can also be applied to browser statistics to obtain a fingerprint of a user.

While click patterns (Nagaraja and Shah, 2019) or drawing symbols (Syukri et al., 1998) have already been used to calculate a fingerprint, there exists also work using mouse statistics for user identification (Ahmed and Traore, 2007). In (Ahmed and Traore, 2007) the recorded 22 subjects in 998 sessions with a session length of 45 seconds. Afterwards, they computed mouse statistics like average mouse velocity, click frequency, and a mouse angle histogram. Similar statistical features were used in (Chu et al., 2013) to detect if the current user is human or a bot. In (Kratky and Chuda, 2018) the mouse statistics are combined with keyboard statistics like average key press duration and key press latency. The same statistics are used in (Solano et al., 2020) for a behavioral login application. The mouse statistics were also combined with the same statistics computed on the gaze of a person during game playing (Kasprowski and Harezlak, 2018). A deep learning approach was proposed in (Hu et al., 2019) where the authors draw the mouse movements between two events (Key press) on the entire scene and feed it into a deep neural network. The disadvantage of this approach is, that it requires a lot of computational resources as well as it also leads to many misclassifications.

We propose to use spatial distributions as well as click distributions. This leads to a reduced data representation which can be effectively used to identify the user. In addition, this data only requires a resource saving machine learning approach to identify the user.

3 METHOD & DATA RECORDING

The data was recorded on two PCs with two different browsers each. On each PC, six recordings per person were made, each of which had a minimum length of five minutes and could be of any length. On each computer, there were two browsers, each of which was used for three recordings. The choice of websites was limited to six and selected with buttons on the top and bottom of the web page. In addition, all sub-pages could be reached as well as other sites could be visited through internal links. This limitation was due to the need of an iframe, which allowed us to keep the recording software running constantly. We have also chosen to use a fixed selection of pages that are the same for all subjects, as it is guaranteed that the subjects receive the same stimulus. This reduces the influence of completely different pages on the gaze and mouse data. Before each recording, the eye tracker software WebGazer (Papoutsaki et al., 2016) was calibrated. During the cal-

ibration, the subject had to gaze at each calibration point and click on each point two times. This information was given to WebGazer (Papoutsaki et al., 2016) as calibration coordinates. After the calibration, the recording started. In total, six people performed the study, which brings the total number of images to 72 ($2\text{browsers} * 2\text{computers} * 3\text{images} * 6\text{people} = 72$).

The collected data is the gaze signal encoded as heatmap, the mouse movements as heatmap and the browser statistics. For the heatmap, we have quantized the data into a 10×10 grid, which is valid for both the eye movement and the mouse movement. In addition, we normalized the sum of the heatmap to one. The collected browser statistics are standard values like webdriver, WebGL, header, language, device memory, etc. according to the FingerprintJS (mis, 2020c). To store the data online we used a local Apache server (Wolfgarten, 2004) with a MySQL database (Greenspan and Bulger, 2001), to which the data was sent via Ajax (Garrett et al., 2005) using Javascript (Goodman, 2007).

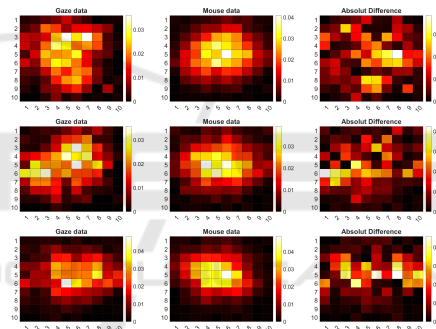


Figure 1: The gaze, mouse, and absolute difference for three users. Each row corresponds to one subject.

Figure 1 shows the normalized gaze and mouse movement data. Each row corresponds to a separate image. Comparing the mouse and gaze data, a clear difference can be seen where both signals have the main focus relatively central. Since we used an iframe for our recordings, we could not use tabs in the browser. Scrolling was also done mainly over the mouse wheel, and the scrollbar was a bit inside the screen. In the third column in Figure 1, which represents the absolute difference, it can be seen that the signals are clearly different. Nevertheless, the signals correlate with each other, which is of course also due to the fact that the heatmap is a quantization and is invariant to time.

Figure 2 shows the distribution as a whisker plot of the correlation coefficient over all recorded data between the gaze and mouse signal. The blue box represents the 75% confidence interval and the red line represents the median. Since all values of the correla-

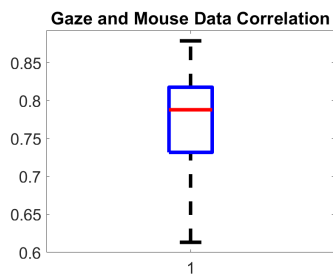


Figure 2: Correlation between the mouse and the gaze data over all samples as whisker plot.

tion coefficient are above 0.5, it can be assumed that there is a very strong relation between the gaze signal and the mouse signal in our recordings. As already mentioned, this is reinforced by the heatmap quantization, the normalization, and the removal of the time dependency in the heatmap.

Larger Scale Study of Mouse Movements: In this study, we recorded the mouse movements of 80 people. Each person made ten recordings and could determine the websites, duration, device (As long as our software was running on it) and also the activity completely freely. With this study, we would like to show that it is also possible to distinguish people on a larger scale based on their mouse movements. The age of the test persons was between 24 and 39. We recorded the mouse movements as well as the clicks with the mouse (left, right and middle mouse button as well as the mouse wheel movements upwards and downwards). The recording was done with a program running in the background so that the persons were completely free in the choice of their activity as well as in the choice of the web pages. There were also no time restrictions for the recordings, so the recording time of our data was between 4 and 25 minutes. As in the first study, we coded each entire recording into a 10×10 heatmap (quantizing the resolution into 10×10 fields) and normalized over their sum. Mouse clicks were also divided by their sum to make this usable as a distribution. In our evaluation, experiments are performed with and without the mouse click distribution.

4 EVALUATION OF THE FIRST SMALL STUDY

For all our evaluations, we performed a 50% to 50% split between training and validation data. We made sure that every user is present in the training data, as well as every computer and browser. This has been done because it is necessary to have seen the user at least once to recognize him. To make the compari-

son to the browser statistics fair, we made sure that every computer and browser is in the training data as well. As a machine learning method, we have chosen bagged decision trees with the standard parameters of Matlab 2020b. The only setting we have made is the number of trees to be trained, which we have set to 50 for all evaluations. We did not perform any data augmentation or other preprocessing. We have chosen this approach because it is the easiest to reproduce. In addition, this work is not about the best possible results, but about the proof of concept of using the mouse as well as gaze data to create a digital fingerprint. The script and data can be viewed in the supplementary material and tested together with Matlab.

Figure 3 shows the results for the use of the individual data sources (browser statistics, mouse, and gaze) separately. The results are displayed as a confusion matrix to view each class separately. Each confusion matrix also has the overall accuracy in the lower right corner. The matrix on the left side was only evaluated with the browser statistics as input. The overall accuracy is exactly at the chance level (16.66%). This shows that the browser statistics, in the case of computers with multiple users, cannot be used effectively to distinguish between the different users. This is because if two users on the same computer in the same browser have the same statistical values. The middle confusion matrix in Figure 3 shows the results achieved with the mouse heatmap. The overall accuracy of 69.4% is significantly above the chance level of 16.66%. Thus, it is clear that the mouse data contains information about the user. Furthermore, this data can be used to distinguish between users who have used the same computer and the same browser. The right matrix in Figure 3 shows the results of the gaze data. These results exceed the results of the mouse data with 80.6% accuracy. This also means that the gaze data can be used to differentiate between users on the same computer, and this even better than any other data source.

Figure 4 shows the results for the combinatorial use of the individual data (browser statistics, mouse, and gaze). Like the individual results, the combinatorial evaluations are also displayed as a confusion matrix. The left matrix shows the results of the combination of browser statistics and the mouse heatmap. This combination is slightly worse than using the mouse heatmap alone (69.4 to 63.9%). This is mainly due to the fact that in our study, the users are equally distributed over all computers and browsers. Normally, that is, that every user has his own computer, except for a few, the browser statistics alone would be very effective and the combination of mouse and

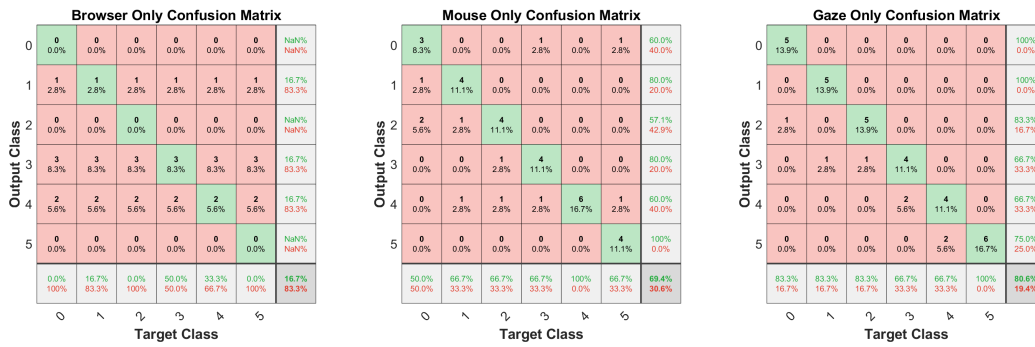


Figure 3: The confusion matrices with the browser statistics, as well as for the gaze and mouse data as input separately. On the bottom right of each confusion matrix, the overall accuracy can be seen.

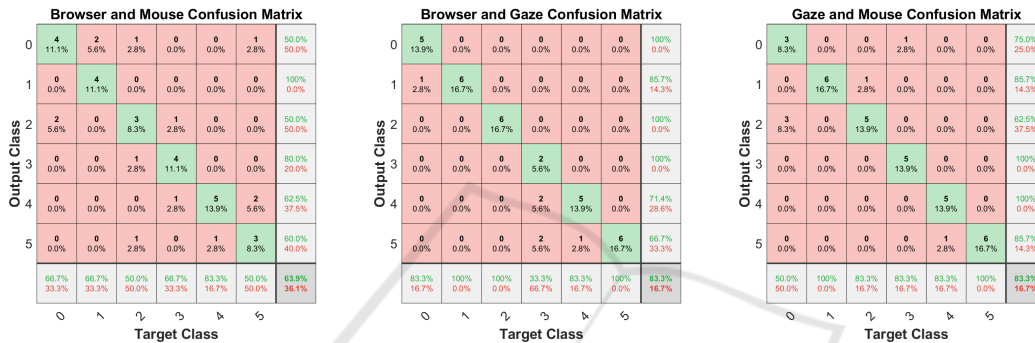


Figure 4: The confusion matrices for the combination evaluations. The left most confusion matrix is the browser and mouse data as input. In the central plot, the results for the browser and gaze data as input are shown. For the right confusion matrix, we combined the gaze and mouse heatmap. On the bottom right of each confusion matrix, the overall accuracy can be seen.

browser statistics would be much better. In the middle matrix of Figure 4 the results of the combination of gaze and browser statistics are shown. Here a slight improvement can be seen (80.6% to 83.3%). This is not very much, but the same applies as for the combination of mouse and browser statistics. Usually the browser statistics is very effective, because most of the users have their own computer, so this combination would be much more effective. The last and right matrix in Figure 4 shows the results of the mouse and gaze heatmaps. Here is only a small improvement to the gaze heatmaps alone (2.7%). One reason for this is that the two heatmaps have a very similar content and therefore correlate strongly (Figure 2). In addition, larger data volumes of significantly more than six users would have to be included in order to be able to finally evaluate this. In general, it can be assumed that this combination is not very effective.

Figure 5 shows the results for the combinatorial use of all data together (browser statistics, mouse, and gaze). As in all previous evaluations, we have used a confusion matrix. The overall result of all data as input is as good as the result when using the gaze data alone (80.6 to 80.6). The individual values of the correct and incorrect classifications in the matrix differ

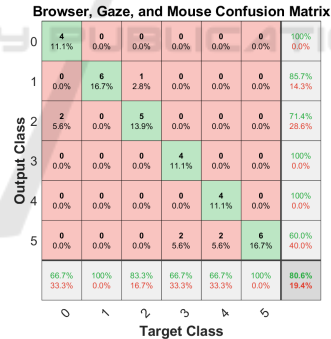


Figure 5: The confusion matrix with the combination of all data sources which are the browser statistics, the gaze heatmap, and the mouse heatmap. On the bottom right, the overall accuracy can be seen.

slightly, but the overall result shows no improvement. Of course, as for all combinatorial analysis with the browser data, the browser data usually works very well and the result is certainly much better, if there are only a few users which share a computer. Also, the combination of the gaze and mouse heatmap may not be optimal, because these data correlate too strongly. However, the gaze heatmap can be replaced with the mouse heatmap. The mouse data have the clear ad-

vantage that they can always be retrieved and do not require calibration.

5 EVALUATION OF THE LARGER SCALE STUDY

Table 1: Shows the average validation results of a 5 folds cross validation. The task for the different classifiers was the classification of the person (80 persons in the dataset) based on a heatmap or a heatmap and click distribution. We used the standard parameters of the Matlab classification learner and compared to a reimplementation of the state-of-the-art mouse statistics (Mouse stats).

Method	Heatmap	Heatmap and clicks	Mouse stats
BaggedTree	90.75%	90.5%	83.5%
Discriminant	98.125%	98.375%	87.125%
KNNEnsem.	94.0%	92.5%	90.0%
CubicSVM	93.5%	93.625%	35.75%
LinearSVM	97.0%	95.25%	86.375%
GaussSVM	91.0%	88.375%	82.25%
QuadricSVM	95.125%	95.375%	41.625%
FineKNN	97.125%	97.75%	87.25%
WeightKNN	97.25%	96.625%	89.625%

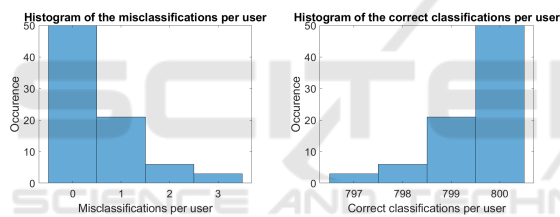


Figure 6: Histogram of the misclassification occurrence per user on the left and the histogram of correct classification occurrence per user on the right for the one vs all evaluation per subject without the click distribution (All confusion matrices for each user are in the supplementary material). This means that each user had to be distinguished from all other users based on a binary classification. We conducted this experiment since it should be closer to the usage as a security mechanism in online banking or similar, where it is about validating the user based on his behavior. For each user, we conducted a 5-fold cross validation and did not balance the dataset (Which can be seen in the confusion matrices based on the numbers of class one and two) nor used any reweighting mechanism. The results for all confusion matrices are from the Matlab 2020b FineKNN with the standard parameters as they are used in the classification learner application. The reason for this histogram evaluation is to show that the approach works for all users.

Table 1 shows the results of person classification based on the heatmap and the heatmap combined with the click distribution. The underlying data are the mouse recordings of the 80 subjects in our second study. We evaluated each method with a 5-fold cross validation and calculated the mean accuracy. As can be seen in Table 1, the Ensemble of subspace dis-

criminant classifiers is the method with the best accuracy (98.125% and 98.375%) second is FineKNN (97.125% and 97.75%). For all procedures, we used the default parameters of Matlab and did not perform a grid search for the optimal parameters. This, together with the scripts and data provided, should make the results easy to reproduce. If we compare the results of the heatmap as well as the results of the heatmap with click distribution, we can clearly see that the click distribution has a rather negative effect in most cases. This is certainly due to the fact that little of an individual’s behavior is reflected in the click behavior, since people are restricted in their clicks and the left mouse button and scrolling are probably the most frequently used functions. This evaluation shows that mouse movements are a very good way to detect users based on their behavior.

Since the evaluation in Table 1 is a detection and identification of one person out of many, this form of use is rather less suitable for security-relevant scenarios such as online banking. Therefore, we have performed further experiments in which the goal is to validate one person out of many. For this, we assigned class one to a single person and class two to all others. In this way, the machine learning algorithms learn to recognize a person based on their behavior and to validate them based on their behavior. Which can basically provide an additional layer of security in online banking or other security related network services.

The evaluations per person as histograms can be seen in Figure 6. Here we performed a 5-fold cross validation for each person and attempted to validate this person against all others using Matlab 2020b’s FineKNN. No balancing or cost function weighting was used for training and evaluation, which means that the results can also be further improved. As can be seen in Figure 6 on the left, there are only rare amounts of misclassifications per user. This is especially true if the histogram of correct classifications (On the right side of Figure 6) is considered.

For the results in Figure 7, we also did a 5-fold cross validation for each individual person. However, here we computed the confusion matrix over all results to compare different machine learning methods and the heatmap in combination with the click distribution for this task. As can be seen in the confusion matrices in Figure 7, FineKNN is by far the best method in terms of the first row of the confusion matrix. This means that it is least likely to classify a valid person as invalid. If we compare the first 6 confusion matrices with the last 6, we see that the click distribution worsens the results, as it does for the person identification (Table 1).

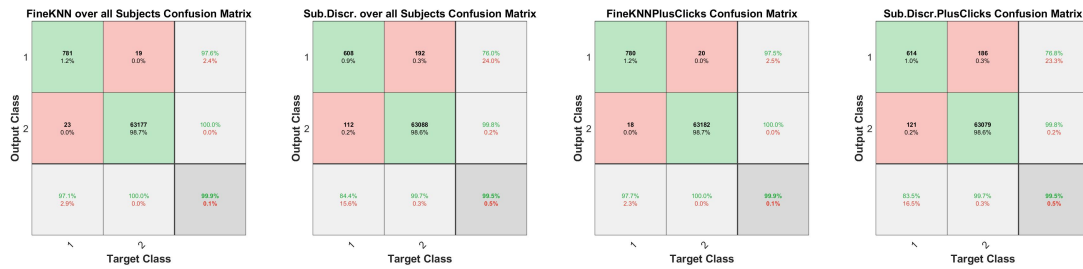


Figure 7: The confusion matrices for different machine learning methods. We conducted the same per user evaluation as in Figure 6 but computed the confusion matrix on all predictions together. This means we did a 5-fold cross-validation for each person vs all other persons and did not rebalance the dataset nor reweight the cost function. This does not show that it works for each user equally good (But this was shown in Figure 6 already). The first two confusion matrices are with the heatmap, and the last two confusion matrices are with the heatmap in combination with the click distribution.

6 CONCLUSION

In this work, we have done a small study and analyzed both gaze and mouse data to use them for a digital fingerprint. In our evaluation, it is clearly shown that these signals can be used individually as well as in combination for fingerprinting. It also shows that in the case of computers used by multiple users, the browser statistic fails and can no longer distinguish the persons. With our data we can confirm, as in previous work, that the gaze and mouse signals are dependent on each other.

ACKNOWLEDGEMENTS

Daniel Weber is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy – EXC number 2064/1 – Project number 390727645

REFERENCES

(2020a). Device tracking add-on for minfraud services - maxmind. <https://dev.maxmind.com/minfraud/device/>.

(2020b). The evolution of hi-def fingerprinting in bot mitigation - distil networks. <https://resources.distilnetworks.com/all-blogposts/device-fingerprinting-solution-botmitigation>.

(2020c). Fingerprintjs. <https://github.com/fingerprintjs/fingerprintjs>.

Ahmed, A. A. E. and Traore, I. (2007). A new biometric technology based on mouse dynamics. *IEEE Transactions on dependable and secure computing*, 4(3):165–179.

Alaca, F. and Van Oorschot, P. C. (2016). Device fingerprinting for augmenting web authentication: classification and analysis of methods. In *Proceedings of the*

32nd annual conference on computer security applications, pages 289–301.

Bursztein, E., Malyshev, A., Pietraszek, T., and Thomas, K. (2016). Picasso: Lightweight device class fingerprinting for web clients. In *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices*, pages 93–102.

Cao, Y., Li, S., Wijmans, E., et al. (2017). (cross-) browser fingerprinting via os and hardware level features. In *NDSS*.

Choubey, J. and Choubey, B. (2013). Secure user authentication in internet banking: a qualitative survey. *International Journal of Innovation, Management and Technology*, 4(2):198.

Chu, Z., Gianvecchio, S., Koehl, A., Wang, H., and Jajodia, S. (2013). Blog or block: Detecting blog bots through behavioral biometrics. *Computer Networks*, 57(3):634–646.

Cole, E. (2011). *Network security bible*, volume 768. John Wiley & Sons.

de Souza, G. B., da Silva Santos, D. F., Pires, R. G., Marana, A. N., and Papa, J. P. (2019). Deep features extraction for robust fingerprint spoofing attack detection. *Journal of Artificial Intelligence and Soft Computing Research*, 9(1):41–49.

Eckersley, P. (2010). How unique is your web browser? In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 1–18. Springer.

Englehardt, S. and Narayanan, A. (2016). Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1388–1401.

Fossen, B. L. and Schweidel, D. A. (2019). Measuring the impact of product placement with brand-related social media conversations and website traffic. *Marketing Science*, 38(3):481–499.

Fuhl, W., Bozkir, E., Hosp, B., Castner, N., Geisler, D., C., T., and Kasneci, E. (2019). Encodji: Encoding gaze data into emoji space for an amusing scanpath classification approach ;). In *Eye Tracking Research and Applications*.

- Fuhl, W., Bozkir, E., and Kasneci, E. (2020). Reinforcement learning for the privacy preservation and manipulation of eye tracking data. *arXiv preprint arXiv:2002.06806*.
- Gabryel, M., Grzanek, K., and Hayashi, Y. (2020). Browser fingerprint coding methods increasing the effectiveness of user identification in the web traffic. *Journal of Artificial Intelligence and Soft Computing Research*, 10(4):243–253.
- Garrett, J. J. et al. (2005). Ajax: A new approach to web applications.
- Gómez-Boix, A., Laperdrix, P., and Baudry, B. (2018). Hiding in the crowd: an analysis of the effectiveness of browser fingerprinting at large scale. In *Proceedings of the 2018 world wide web conference*, pages 309–318.
- Goodman, D. (2007). *JavaScript bible*. John Wiley & Sons.
- Greenspan, J. and Bulger, B. (2001). *MySQL/PHP database applications*. John Wiley & Sons, Inc.
- Guo, Q. and Agichtein, E. (2010). Towards predicting web searcher gaze position from mouse movements. In *CHI'10 Extended Abstracts on Human Factors in Computing Systems*, pages 3601–3606.
- Holland, C. and Komogortsev, O. V. (2011). Biometric identification via eye movement scanpaths in reading. In *2011 International joint conference on biometrics (IJCB)*, pages 1–8. IEEE.
- Hu, T., Niu, W., Zhang, X., Liu, X., Lu, J., and Liu, Y. (2019). An insider threat detection approach based on mouse dynamics and deep learning. *Security and Communication Networks*, 2019.
- Juels, A., Jakobsson, M., and Jagatic, T. N. (2006). Cache cookies for browser authentication. In *2006 IEEE Symposium on Security and Privacy (S&P'06)*, pages 5–pp. IEEE.
- Kasprowski, P. and Harezlak, K. (2018). Biometric identification using gaze and mouse dynamics during game playing. In *International Conference: Beyond Databases, Architectures and Structures*, pages 494–504. Springer.
- Kobusińska, A., Pawluczuk, K., and Brzeziński, J. (2018). Big data fingerprinting information analytics for sustainability. *Future Generation Computer Systems*, 86:1321–1337.
- Kratky, P. and Chuda, D. (2018). Recognition of web users with the aid of biometric user model. *Journal of Intelligent Information Systems*, 51(3):621–646.
- Laperdrix, P., Bielova, N., Baudry, B., and Avoine, G. (2020). Browser fingerprinting: a survey. *ACM Transactions on the Web (TWEB)*, 14(2):1–33.
- Lee, Y. S., Kim, N. H., Lim, H., Jo, H., and Lee, H. J. (2010). Online banking authentication system using mobile-otp with qr-code. In *5th International Conference on Computer Sciences and Convergence Information Technology*, pages 644–648. IEEE.
- Liebling, D. J. and Dumais, S. T. (2014). Gaze and mouse coordination in everyday work. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: adjunct publication*, pages 1141–1150.
- Ludwig, S. A. (2019). Applying a neural network ensemble to intrusion detection. *Journal of Artificial Intelligence and Soft Computing Research*, 9(3):177–188.
- Mayer, J. R. (2009). Any person... a pamphleteer?: Internet anonymity in the age of web 2.0. *Undergraduate Senior Thesis, Princeton University*, page 85.
- Nagaraja, S. and Shah, R. (2019). Clicktok: click fraud detection using traffic analysis. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 105–116.
- Navalpakkam, V., Jentzsch, L., Sayres, R., Ravi, S., Ahmed, A., and Smola, A. (2013). Measurement and modeling of eye-mouse behavior in the presence of nonlinear page layouts. In *Proceedings of the 22nd international conference on World Wide Web*, pages 953–964.
- Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., and Hays, J. (2016). Webgazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence-IJCAI 2016*.
- Renjith, S. (2018). Detection of fraudulent sellers in online marketplaces using support vector machine approach. *arXiv preprint arXiv:1805.00464*.
- Salakhutdinov, R. and Hinton, G. (2009). Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978.
- Shamdasani, P. N., Stanaland, A. J., and Tan, J. (2001). Location, location, location: Insights for advertising placement on the web. *Journal of Advertising Research*, 41(4):7–21.
- Solano, J., Tengana, L., Castelblanco, A., Rivera, E., Lopez, C., and Ochoa, M. (2020). A few-shot practical behavioral biometrics model for login authentication in web applications. In *NDSS Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb'20)*.
- Syukri, A. F., Okamoto, E., and Mambo, M. (1998). A user identification system using signature written with mouse. In *Australasian Conference on Information Security and Privacy*, pages 403–414. Springer.
- Tucker, C. E. (2014). Social networks, personalized advertising, and privacy controls. *Journal of marketing research*, 51(5):546–562.
- Vastel, A., Rudametkin, W., Rouvoy, R., and Blanc, X. (2020). Fp-crawlers: Studying the resilience of browser fingerprinting to block crawlers. In *NDSS Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb'20)*.
- Wolfgarten, S. (2004). *Apache Webserver 2: Installation, Konfiguration, Programmierung*. Pearson Deutschland GmbH.
- Zhang, X., Han, Y., Xu, W., and Wang, Q. (2019). Hoba: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture. *Information Sciences*.