Assessing User Behavior by Mouse Movements

Jennifer Jorina Matthiesen(✉) and Ulf Brefeld

Machine Learning Group, Leuphana University of Lüneburg, Lüneburg, Germany
jennifer.matthiesen@stud.leuphana.de, ulf.brefeld@leuphana.de

Abstract. In this working paper, we study user identification via mouse movement. Instead of treating the problem as a multi-class classification task, we cast user identification as a one-class problem and propose to learn an individual model for every user. Preliminary empirical results show that our approach works for some but not all users. We report on lessons learned.

Keywords: Mouse movement · User identification · User behavior

1 Introduction

User identification is not only key to privacy and security but also offers a way to personalize user experiences, e.g., by displaying user-specific content. Apart from biometric user identification, a non-intrusive alternative is offered by user behavior. In contrast to physical traits, behavioral-based authentication allows for continuous (re-)identification during user sessions. In this context, particularly mouse movements is of interest, as it does not require additional hardware and allows implicit and non-inversive measurements of behavioral biometrics [10,11,16,17].

Similar to gestures in human communication, the dynamics of the pointing device in human-computer interaction are unique and can deliver valuable and deterministic information about the user [2,9,11,16,17]. However, the question raises how such a system could be reasonably build. Traditionally, a multi-class classification approach suggests itself: every user is identified with a class and a classifier chooses the most likely user among the candidates. While such an approach may work for hardly changing environments, dynamic scenarios with many new and deleted users require frequent retraining of the classifier. For a large user base with many sessions per day, this could quickly become infeasible.

By contrast, we treat user identification as an anomaly detection problem [12,13,15] and propose to learn a model of normality for every user. The idea is as follows: As long as the user interacts with the system, the corresponding model correctly identifies the user. If a third user takes over, the identification fails and the model considers the third party as an anomaly and may shut down critical applications and data access points. Maintaining a multitude of these models is simple. Once a user logs in, the right model is retrieved and used until the end of the session.

C. Stephanidis and M. Antona (Eds.): HCII 2020, CCIS 1224, pp. 68–75, 2020.
https://doi.org/10.1007/978-3-030-50726-8_9

Retraining can be trivially parallelized for all users, new user models are integrated by training a single new model, and a deletion of a user simply deletes the corresponding model without any side effects for other models.

Our contributions are as follows: (i) We cast user identification as an anomaly detection problem, where user profiles are learned in a rich (non-linear) feature space spanned by a set of automatically derived features [13,15] and in a deep neural architecture [12]. (ii) We evaluate the impact of splitting sessions into sequences including pause-based and an equal number of data points splits. (iii) We report on lessons learned that may shed light on future research in this area.

2 Related Work

Mouse movement has been investigated in the context of user authentication [4,5,14] and behavioral analyses [2,3,10,16]; a great deal of these publications rely on hand-engineered features [2,3,5,8,14,16] though. User identification based on biometrics extracted from mouse behavior has been first introduced by Gamboa & Fred [6]. They proposed a number of features and split the session into single sequences based on mouse clicks. Features are subsequently reduced by greedy search and fed into a sequential classifier. Feher et al. [5] introduce a hierarchical structure of mouse features, proposing in total 66 features. With these features, a random forest classifier is trained using 30 actions for verification. Recently, Chong et al. [4] investigate different architectures for user authentication using mouse data. However, their approach requires to retain the full model with samples of all users, when a new user is added.

3 Algorithms

Perhaps the most prominent one-class-classifier is the **One-Class Support-Vector-Machine** (OC-SVM) [13]. Its objective is to find the max-margin hyperplane that separates the origin from the data, where the latter is mapped by a function ϕ into a (possibly nonlinear) feature space spanned by $\phi : \mathcal{X} \mapsto \mathcal{F}$. Given a training set $\mathcal{D} = \{x_1, ..., x_n\}$ with $x_i \in \mathcal{X}$, the primal problem of the OC-SVM can be written as

$$\min_{\boldsymbol{\omega},\rho,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{\omega}\|^2 - \rho + \frac{1}{\nu n}\mathbf{1}^\top \xi \quad \text{s.t.} \; \forall \; i : \; \boldsymbol{\omega}^\top \phi(\boldsymbol{x}_i) \geq \rho - \xi_i \; \wedge \; \xi_i \geq 0$$

where ρ is the distance of the hyperplane to the origin and acts as a threshold such that a new instance $\boldsymbol{x}$ is considered anomalous (not belonging to the class that is represented by data $\mathcal{D}$) if $f(\boldsymbol{x}) = \boldsymbol{\omega}^\top \phi(\boldsymbol{x}) - \rho < 0$.

The **Support Vector Data Description** (SVDD) [15] is similar to the OC-SVM, but uses a hypersphere as a model of normality. The objective of the SVDD is to find the smallest hypersphere, given by radius $R > 0$ and center $c \in \mathcal{F}$, which encloses the majority of the data in feature space. The primal optimization problem is given by

$$\min_{\boldsymbol{c},R,\boldsymbol{\xi}} \quad R^2 + \frac{1}{\nu n}\mathbf{1}^\top \xi \quad \text{s.t.} \; \forall \; i : \; \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 \leq R^2 + \xi_i \; \wedge \; \xi_i \geq 0.$$

New points are considered anomalous if they lie outside of the hyperball, that is, if $\|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2 > R^2$.

Recently, [12] presented a deep variant of the SVDD. An autoencoder is used for dimensionality reduction while a second part of the network minimizes the volume of the data-enclosing hypersphere. The objective of the **Deep SVDD** [12] is given by

$$\min_{\boldsymbol{c},R,\mathcal{W}} \quad R^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \max\{0, \|\phi(\boldsymbol{x}_i;\mathcal{W}) - \boldsymbol{c}\|^2 - R^2\} + \frac{\lambda}{2}\sum_{l=1}^{L} \|\mathcal{W}^l\|^2 \quad \text{s.t. } R > 0$$

The second term penalizes points lying outside the sphere analogously to the traditional SVDD.

4 Empirical Study

We use data from the Balabit Mouse Dynamic Challenge[1] that comprises sessions from ten different users. The training data encompasses five to seven longer sessions for each user while the test set contains multiple smaller sessions. The test set contains also out-of-sample users not present in the training set as well as sessions from anonymous attackers. The latter are simulated by mixing sessions from other users into the test session of a third user. Table 1 summarizes the data.

Table 1. Overview of Balabit Mouse Dynamics Challenge dataset

User	Training			Test				
	files	min_length	max_length	legal	illegal	sessions	min_length	max_length
7	7	43484	**83091**	36	37	73	164	6966
9	7	54418	72732	23	43	66	141	10991
12	7	29722	48244	56	49	105	127	8086
15	6	16971	44015	45	70	115	119	5656
16	6	28428	53816	68	38	106	**114**	3104
20	7	31441	60087	30	20	50	146	**12672**
21	7	15343	21465	37	22	59	154	2170
23	6	17127	28435	38	33	71	157	4706
29	7	**13640**	32601	43	20	63	134	5207
35	5	16901	23107	35	73	108	114	3771
	65			411	405	816		

[1] https://github.com/balabit/Mouse-Dynamics-Challenge.

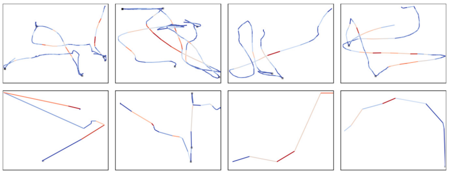

Fig. 1. Sequences produced by the TDS-method, using the 98% quantile of the overall pauses as a splitting criterion. The first row shows legal, while the second row shows illegal sessions of user 7.

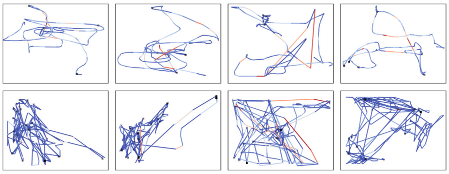

Fig. 2. Sequences produced by the EDPS-method, using 200 data points as a splitting criterion. The first row shows legal, while the second row shows illegal sessions of user 7.

4.1 Session Splitting

We split the mouse movement within a session into short sequences. We investigate two different splitting criteria, the first Time Difference Split (TDS) and Equal Number Of Data Points Split (EDPS).

The former splits the session by time differences between two consecutive mouse coordinates. The pauses made by the user during the interaction with the system are an active field of mouse movement research [5,6,16]. Our approach is similar to [4], but instead of setting a hyper-parameter for the time difference splitting criterion, we determine the parameter based on the users' mouse data using quantiles. We study the effect of splitting mouse movements at 95%, 98% and 99% quantiles. This leads to a unique splitting criterion for every user, see Fig. 1.

EDPS splits mouse data into sequences by using a fixed number of data points. We investigating different lengths of sequences ($m \in \{50, 100, 200, 500\}$). Using a fixed number of data points as a splitting criterion ensures that the session is separated and provides sequences of the same length, see Fig. 2.

The splitted sequences are cleaned and the resulting logs contain the following variables: timestamps, (x, y) coordinates, mouse buttons (left, right, scroll) and the action type (move, pressed, released, drag). Since the velocity of a scroll is not given we discard the related actions and ignore scroll operations entirely. We compare the 65 features from [5] with additional 12 features described in the Appendix (Table 2). All features are normalized.

We evaluate area under the curve (AUC) and the equal error rate (EER). The latter is identical to the intersection of the false acceptance rate (FAR) and the false rejection rate (FRR). To not clutter the evaluation part unnecessarily, we report only results for TDS using the 99% quantile and EDPS with length 100 that worked best over all tested parameter settings.

We compare OC-SVM, SVDD, and Deep SVDD and also include a vanilla SVM trained in a one-versus-rest manner, denoted by OvR-SVM for interpretability. The results are shown in Fig. 3 for TDS-99% and Fig. 4 for EDPS-50.

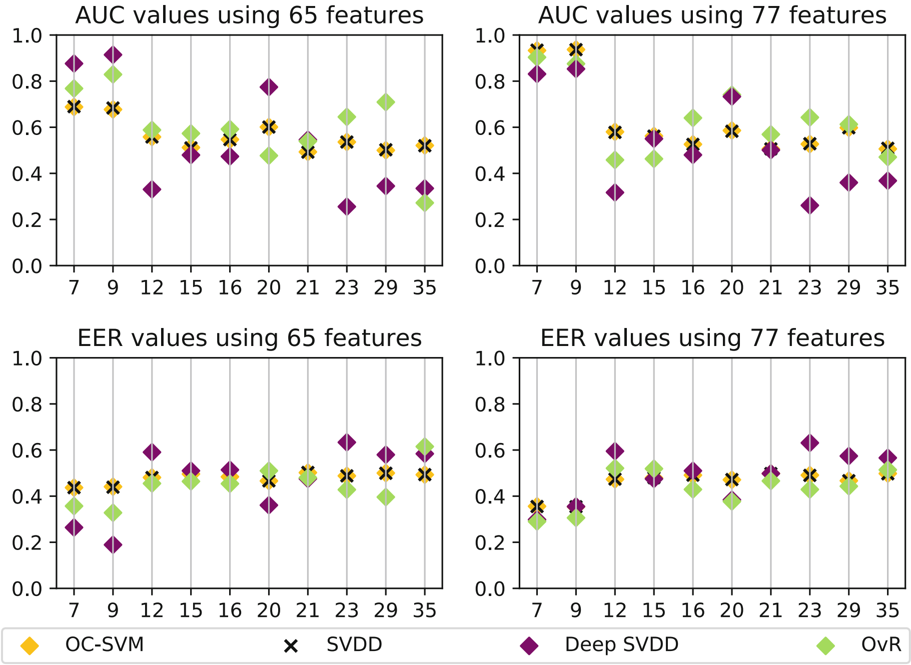

Fig. 3. Results for TDS-99%

Unsurprisingly, the OvR-SVM clearly outperforms the one-class approaches. However, OvR-SVM also uses more information by including unified data from all other users as the negative class in the training process. Thus, OvR-SVM shows that there is room for improvement for the methods of interest, but, by construction, poses a solution that is clearly inappropriate in many practical scenarios. Also unsurprisingly, OC-SVM and SVDD perform equivalent throughout the experiment; for certain normalized feature representation, their objective functions become identical and provide naturally the same solution. The Deep SVDD performs well for user 7, 9 and 20 for sequences derives by the EDPS- as well as the TDS-method on 65 and 77 respectively. This finding gives rise to two conjectures: The first is that some users can, in general, be identified by their mouse movement as was also shown e.g., in [1,4–6,14]. And second, that perhaps the feature representation was simply not the right one for the other users. Thus, it can be hypothesised that the features learned for the authentication process have to be individualized so that the detection performance is maximized (see e.g., [7]).

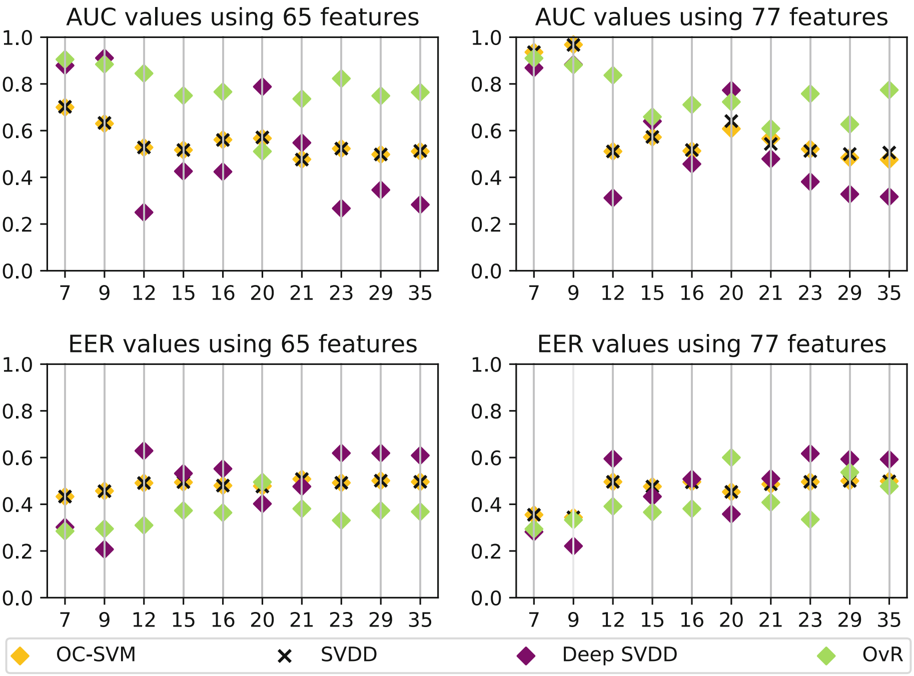

Fig. 4. Results for EDPS-100

5 Conclusions

We studied user identification by mouse movements. Conceptually, we interpreted the problem setting as an anomaly detection problem and evaluated traditional (OC-SVM, SVDD, OvR) and recent (DeepSVDD) methods. Preliminary empirical results showed that some users can actually be identified solely based on their mouse movement. This finding however does not hold for most of the users. Our lessons learned is twofold: (i) We conjecture that mouse behaviour is idiosyncratic, which is in line with other studies [1,4–6,14], and (ii) that we might be able to improve user identification by tailoring (learning) an individual feature representation for every user.

A Additional Features

Table 2. List of additional features

Feature name	Description	Formal definition
Traveled distance	The sum of the distance between points	$\delta s = \sqrt{\sum_{i=1}^{n} (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$
Number of data points	Just used in TDS-method	n
Pauses length	min, max, $mean$, σ $(max - min)$	δt
Number of pauses	–	$\sum_{i=1}^{n} p_i \ where \ p_i = \begin{cases} 1, & \delta t_i > threshold \\ 0, & \text{otherwise} \end{cases}$
Number of clicks	–	$\sum_{i=1}^{n} c_i$
Dispersal x	–	$dis_x = \sqrt{(x_{max} - x_{min})^2}$
Dispersal y	–	$dis_y = \sqrt{(y_{max} - y_{min})^2}$
Dispersal	–	$dis = dis_y * dis_x$

References

1. Antal, M., Egyed-Zsigmond, E.: Intrusion detection using mouse dynamics. IET Biometrics **8**, 285–294 (2019)
2. Arapakis, I., Lalmas, M., Valkanas, G.: Understanding within-content engagement through pattern analysis of mouse gestures. In: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, pp. 1439–1448. ACM (2014)
3. Arapakis, I., Leiva, L.A.: Predicting user engagement with direct displays using mouse cursor information. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Inform. Retrieval, pp. 599–608. ACM (2016)
4. Chong, P., Elovici, Y., Binder, A.: User authentication based on mouse dynamics using deep neural networks: a comprehensive study. IEEE Trans. Inf. Forensics Secur. **15**, 1086–1101 (2020)
5. Feher, C., Elovici, Y., Moskovitch, R., Rokach, L., Schclar, A.: User identity verification via mouse dynamics. Inf. Sci. **201**, 19–36 (2012)
6. Gamboa, H., Fred, A.: A behavioral biometric system based on human-computer interaction. In: Jain, A.K., Ratha, N.K. (eds.) Biometric Technology for Human Identification, vol. 5404, pp. 381–392. SPIE (2004)

7. Kloft, M., Brefeld, U., Sonnenburg, S., Zien, A.: Lp-norm multiple kernel learning. J. Mach. Learn. Res. **12**, 953–997 (2011)
8. Lagun, D., Ageev, M., Guo, Q., Agichtein, E.: Discovering common motifs in cursor movement data for improving web search. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 183–192. ACM (2014)
9. McNeill, D.: Hand and Mind: What Gestures Reveal About Thought. University of Chicago Press, Chicago (1992)
10. Mueller, F., Lockerd, A.: Cheese: tracking mouse movement activity on websites, a tool for user modeling. In: CHI 2001 extended abstracts on Human factors, April 2001
11. Navalpakkam, V., Churchill, E.: Mouse tracking: measuring and predicting users' experience of web-based content. In: Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems - CHI 2012 (2012)
12. Ruff, L., et al.: Deep one-class classification. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, 10–15 July 2018, vol. 80, pp. 4393–4402. PMLR (2018)
13. Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. In: Neural Computing, vol. 13, p. 1443–1471. MIT Press, Cambridge, MA, USA (Jul 2001)
14. Shen, C., Cai, Z., Guan, X., Du, Y., Maxion, R.A.: User authentication through mouse dynamics. IEEE Trans. Inf. Forensics Secur. **8**, 16–30 (2013)
15. Tax, D., Duin, R.: Support vector data description. Mach. Learn. **54**, 45–66 (2004). https://doi.org/10.1023/B:MACH.0000008084.60811.49
16. Tzafilkou, K., Protogeros, N.: Mouse behavioral patterns and keystroke dynamics in end-user development: what can they tell us about users' behavioral attributes? Comput. Hum. Behav. **83**, 288–305 (2018)
17. Zimmermann, P., Guttormsen, S., Danuser, B., Gomez, P.: Affective computing - measuring mood with mouse and keyboard. Int. J. Occup. Saf. Ergon. **9**, 539–51 (2003)