Journal of Cybersecurity and Privacy

MDPI

Article

ReMouse Dataset: On the Efficacy of Measuring the Similarity of Human-Generated Trajectories for the Detection of Session-Replay Bots

Shadi Sadeghpour * and Natalija Vlajic

Department of Electrical Engineering and Computer Science, York University, Toronto, ON M3J 1P3, Canada
* Correspondence: shadisa@cse.yorku.ca

Abstract: Session-replay bots are believed to be the latest and most sophisticated generation of web bots, and they are also very difficult to defend against. Combating session-replay bots is particularly challenging in online domains that are repeatedly visited by the same genuine human user(s) in the same or similar ways—such as news, banking or gaming sites. In such domains, it is difficult to determine whether two look-alike sessions are produced by the same human user or if these sessions are just bot-generated session replays. Unfortunately, to date, only a handful of research studies have looked at the problem of session-replay bots, with many related questions still waiting to be addressed. The main contributions of this paper are two-fold: (1) We introduce and provide to the public a novel real-world mouse dynamics dataset named ReMouse. The ReMouse dataset is collected in a guided environment, and, unlike other publicly available mouse dynamics datasets, it contains repeat sessions generated by the same human user(s). As such, the ReMouse dataset is the first of its kind and is of particular relevance for studies on the development of effective defenses against session-replay bots. (2) Our own analysis of ReMouse dataset using statistical and advanced ML-based methods (including deep and unsupervised neural learning) shows that two different human users cannot generate the same or similar-looking sessions when performing the same or a similar online task; furthermore, even the (repeat) sessions generated by the same human user are sufficiently distinguishable from one another.

Keywords: behavioral biometrics; mouse dynamics; feature learning; convolutional neural network; clustering algorithms

Citation: Sadeghpour, S.; Vlajic, N. ReMouse Dataset: On the Efficacy of Measuring the Similarity of Human-Generated Trajectories for the Detection of Session-Replay Bots. *J. Cybersecur. Priv.* **2023**, *3*, 95–117. https://doi.org/10.3390/jcp3010007

Academic Editors: Giorgio Giacinto and Phil Legg

Received: 12 January 2023
Revised: 22 February 2023
Accepted: 27 February 2023
Published: 2 March 2023

1. Introduction

Behavioral biometrics measure and analyze user interactions in the online domain so as to recognize or verify a person's unique identity, with the ultimate goal of providing an imperceptible layer of security to systems and applications [1]. The best-known forms of behavioral biometrics involve the monitoring and analysis of the following modalities: mouse cursor movement, keystroke or voice dynamics, the appearance and speed of signing, etc. The main advantages of mouse movement analysis relative to the other forms of behavioral biometrics include: (a) mouse movement can be monitored in a manner that is entirely unobtrusive for the end user; (b) monitoring of mouse movement does not require the use of additional hardware or software and thus does not incur additional cost; (c) from the perspective of user privacy, sharing mouse dynamics data is far less problematic than sharing keystrokes, signatures or voice data [2]; (d) mouse movement has already proven to be effective, not only in the identification or authentication of end users but also in the process of determining users' age and gender [3], as well as their emotions and work productivity [4].

A number of previous studies on mouse dynamics have looked at the importance of different mouse movement characteristics for the purpose of user identification/authentication,

such as hesitation patterns, random and straight movements, etc. [5]. Some of these studies have also looked at the use of different machine learning methods in user identification/authentication systems; however, they often rely only on a limited number of handpicked features extracted from their respective mouse movement datasets. To avoid the pitfalls of manual feature extraction processes, in this study we propose to tackle the problem of mouse trajectory classification by using a deep neural network (convolutional neural network) that utilizes all of the raw mouse movement data. That is, instead of handpicking the most important features for a set of mouse movement trajectories, we let the convolution neural network identify these features in an unsupervised manner. Furthermore, we investigate the use of mouse movement analysis in another important application area—malicious web-bot detection. Malicious web bots are known to pose a significant threat to the entire Internet community. One particularly challenging form of malicious bot are the bots capable of impersonating human behavior in terms of mouse movement. The latest generation of such human-mimicking malicious bots are synthesized by means of 'session replays' [6–8]. That is, these bots programmatically replay a browsing session, including the mouse movement trajectory, that was previously executed (and recorded) by a genuine human visitor to a target/victim web site. The specific goal of this study is to offer a better insight into: (a) the statistical similarities and differences between browsing sessions (mouse movement trajectories) generated by different genuine users on the same target web page; (b) the statistical similarities and differences between browsing sessions (mouse movement trajectories) repeated by the same genuine user on the same target web page. We believe that a better understanding of these similarities and differences is of critical importance for the creation of more effective techniques of malicious bot detection—in particular the detection of session-replay bots—which in turn can ensure a safer Internet for everyone.

The specific contributions of the research work presented in this paper can be summarized as follows: (i) We developed an interactive web platform capable of collecting a number of different mouse movement actions and features, including trajectory, point-click, drag-and-drop, velocity, etc. The platform has been deployed on MTurk (https://www.mturk.com/, accessed on 25 February 2023) and has allowed us to collect mouse movement data from several hundred genuine human users (i.e., participants) while repeating the same/similar online task. We named this dataset ReMouse and are making it available to the research community on IEEE DataPort [9]. (ii) We conducted statistical and ML-based analyses of the ReMouse dataset. The results of this analysis have shown that all mouse dynamics sessions coming from the same genuine human user are relatively different from each other and that it is highly unlikely that different genuine human users produce 'same-looking' sessions when completing the same/similar online task.

To the best of our knowledge, the ReMouse dataset is the first publicly available mouse dynamics dataset with repeat sessions generated by the same human user(s). As such, this dataset can be a very valuable resource for any future research dealing with the problem of session-replay bots, which are currently known to be the most advanced form of web bots on the Internet. In this work, we make the first step towards the ReMouse dataset analysis using statistical and advanced ML-based methods, including deep and unsupervised neural learning. Given the fact that no prior research on the topic of repeat sessions and/or session-replay bots has been conducted (i.e., that is available in the literature), we needed to develop an entirely new research methodology. With this manuscript, we not only try to close the current research and literature gap, we also highlight the need for further development and hope to inspire other researchers to work alongside us on this important area of study.

The remainder of this paper is organized as follows: In Section 2, we provide an overview of previous relevant works on the use of mouse dynamics for the purpose of user authentication and bot detection, as well as an overview the existing publicly available mouse dynamics datasets, including our novel ReMouse dataset. In Section 3, we introduce the web platform that has been used to collect the ReMouse dataset. In Section 4, we present

the results of our analysis of the ReMouse dataset using statistical analysis techniques, while in Sections 5 and 6, we summarize our approach and main findings obtained on the ReMouse dataset using advanced ML techniques. Finally, conclusions and directions for future work are presented in Section 7.

2. Related Work

Understanding users' behavior on one or a set of related web pages, including the usage of mouse cursors, has been essential in many application domains, including educational technology, web analytics, e-commerce, digital advertising, and especially bot detection and user authentication [10,11]. To date, a substantial number of published works has looked at the importance of mouse dynamics from a number of different research perspectives. In this section, we provide a survey of a subset of works which are more closely related to the topic of our own research. In particular, we provide an overview of published works that have studied mouse dynamics in the context of user authentication and bot detection. We also give an overview of several publicly available mouse dynamics datasets.

2.1. Mouse Dynamics for User Authentication

A number of research works have proven the general usefulness of mouse dynamics in the domain of user authentication. Some of these works have also turned to the use of machine learning as a promising approach to increasing the accuracy of mouse-movement-based authentication.

In [12], the authors have provided a comprehensive study on the use of several different deep learning architectures, i.e., 1D-CNN (convolutional neural network), 2D-CNN, LSTM (long short-term memory) and a hybrid CNN-LSTM in biometric-based authentication systems deploying mouse dynamics data. In particular, the authors have combined convolutional layers with LSTM layers to build a hybrid neural network capable of modeling temporal sequences on a larger but fixed time scale. Another deep learning approach has been proposed in [13] to address the problem of biometric-based user authentication in systems with an insider threat. Specifically, to preserve the mouse movement features of each individual user, a unique mapping method was developed to map all the basic actions, such as move, click, drag, scroll and stay, into images. The obtained (images) dataset was then used to train seven-layer CNN classification models.

An authentication system based on a weighted multi-classifier voting technique and deploying different mouse movement operations (such as movement direction and elapsed time) has been described in [14]. In [15], the authors have applied a semi-supervised learning method using a novel feature extraction technique for authentication via mouse dynamics. The authors of [16] have introduced a user authentication system comprising two components named 'enrollment', responsible for feature learning, and 'verification', which performs the actual authentication. The authors have employed an FCN (fully convolutional neural network) for feature learning and an OCSVM (one-class support vector machine) for authentication.

The use of the Random Forest algorithm for the purpose of user authentication has been studied in [17]. To predict/determine one's identity, this study suggests using approximately 1000 mouse actions (60 min of the user's active mouse movements on average) to train the model. The findings of this study imply that mouse dynamics should be considered as an additional security service in the systems, not a single verification indicator.

In [18], the researchers have improved the results of user authentication based on mouse dynamics by replacing the raw coordinates with directional velocities. Finally, the effectiveness of using ensemble learning and frequency-domain representations of mouse dynamics for continuous authentication tasks have been studied in [19].

2.2. Mouse Dynamics for Bot Detection

To date, the use of mouse movement analysis in another important application area—malicious web-bot detection—has been investigated by several researchers. Acien et al. [20]

have presented a bot detector called BeCAPTCHA-Mouse, which is trained on data generated by the neuromotor modeling of mouse dynamics and is claimed to be capable of detecting highly realistic bot trajectories. To detect web bots, Iliou et al. [21] have proposed a framework that combines two web-bot detection modules: a web-logs detection module and a mouse movement detection module. Each module has its own classifier. The fundamental idea of the proposed approach is to capture the different temporal properties of web logs and mouse movements, plus the spatial properties of mouse movements, with the ultimate goal of creating a more robust detection framework that would be hard to evade.

Other researchers have proved the usefulness of mouse dynamics in detecting malicious bots by employing a deep neural network approach [22]; C4.5 algorithm [8]; a combined model of unsupervised and supervised ML techniques, including the K-Nearest-Neighbors algorithm and naïve Bayes classifier [23], a classification algorithm based on distance measures adapted from the Kolmogorov–Smirnov non-parametric test [24] and sequence learning [25]. Importantly, in [26], the authors have proposed a new web forensic framework for bot crime investigations. The framework is based on four different types of human behavioral patterns (timing, movement, pressure and error) to provide evidence of bad bot activity on web applications.

Although there exists a broad list of machine learning algorithms and data mining techniques that have been applied to the problem of bot detection, the question/problem of advanced session-replay web bots remains largely unanswered. According to our knowledge, the only two research studies that have tackled the problem of session replays and have attempted to build adequate ML-based countermeasures are [8,27]. However, the focus of [27] is on session replays in the context of user authentication (and not malicious web bots), while the results of [8] are based on a proprietary dataset involving blog bots (one very narrow subcategory of web bots). Moreover, a common drawback of both studies is that they omit to consider the possibility of web-sites (i.e., online services) in which genuine human users end up generating similar/repeat sessions, as in the case of news, banking or gaming web-sites.

2.3. Mouse Dynamics Datasets

In terms of the actual mouse movement datasets analyzed in their studies, different researchers have employed different approaches to acquiring human-generated mouse trajectories. They have either used existing publicly available datasets (e.g., [17,28–32]) or they have collected their own. In general, there are two different approaches to collecting a mouse movement dataset: (1) by creating a 'guided environment', where the users are asked to perform a specific (same) task with the mouse, or (2) by creating a 'non-guided environment', where users are not guided (i.e., instructed) on how to perform a particular task [33].

Some of the most commonly studied publicly available mouse movement datasets include: Balabit [28], Bogazici [29], the Attentive Cursor dataset [30], SapiMouse [31], Chao Shen [32] and DFL [17]. The following provides a brief description of each dataset.

2.3.1. Balabit Dataset

Published in 2016, the Balabit dataset falls in the category of 'non-guided environment' datasets and includes mouse pointer positioning and timing information for 10 users working over remote desktop clients connected to a remote server. During data collection, users were asked to perform their regular daily activities. Mouse events were stored in tuples containing the following data: timestamp, pressed button, mouse state and mouse pointer coordinates. The primary purpose of collecting the Balabit dataset was to learn how the involved users utilize their mouse so as to be able to protect them from unauthorized usage of their accounts. Both training and test data are presented as sessions in the dataset; however, the test sessions are much shorter than the training sessions.

2.3.2. Bogazici Mouse Dynamics Dataset

The Bogazici dataset [29], published in 2021, also falls into the category of 'non-guided environment' datasets and comprises mouse usage behavior patterns of 24 users gathered over a one-month period. The data collection participants were selected from different positions in a software company in order to acquire different patterns of user behavior while interacting with different programs and tools in the office environment. Each user's machine was loaded with a specially designed program that would launch at startup and would collect the user's mouse movements without being tied to a specific task and without preventing the user from performing their regular daily activities. The specific information contained in the dataset includes mouse action type, timestamp, spatial coordinates, button, state and application window name. The dataset was collected for the purpose of training several neural network and deep learning models, which were then deployed to identify/verify the involved users.

2.3.3. The Attentive Cursor Dataset

This is a large-scale 'guided environment' dataset of mouse cursor movements during a web search task, and the set was collected in 2020 for the purposes of inferring a user's attention and demographic information. Nearly 3000 participants were recruited from the FIGURE EIGHT (https://www.figure-eight.com, accessed on 25 February 2023) crowdsourcing platform. Using an injected custom JavaScript code, the authors captured the real-world behavior of individuals completing a transactional web search task. The captured information includes the following: mouse cursor position, timestamp, event name, XPath of the DOM element related to the event and the DOM element attributes (if any).

2.3.4. SapiMouse Dataset

The dataset was collected at Sapientia University in 2020 and also falls into the category of 'guided environment' datasets. It contains mouse dynamics data from 120 subjects (92 males and 28 females between 18 and 53 years of age). Using a JavaScript web application running on the user's computer, mouse movements were sampled by an event-driven sampling technique. The participants were asked to perform four different actions, and each was associated with geometrical shapes in a web page, including right and left clicks and drag and drop actions. In the dataset, two files were associated with each participant, with each file corresponding to one- and three-minute-long sessions, respectively. Individual lines in the two files capture information pertaining to one mouse event, such as mouse cursor position, button type, event type (move, drag, press or release) and respective timestamp. The authors have presented user authentication results obtained on this dataset in [31].

2.3.5. Chao Shen Dataset

This 'non-guided environment' dataset was collected in 2017 and consists of mouse dynamics information pertaining to 28 users, with each user completing at least 30 separate data sessions over a two-month period. Each session consisted of about thirty minutes of the respective user's mouse activity. In the dataset, each mouse operation was represented as a tuple of multi-attributes (action type, application type, screen area and window position) and their respective timestamps. The dataset was collected for the purpose of continuous user authentication.

2.3.6. DFL Dataset

This dataset was collected in 2018 from 21 participants in a non-guided environment. The participants were asked to install a background service on their computers (which collected their mouse activity data) and perform their daily activities. The dataset contains the following information about the users' mouse activities: timestamp, button (left, right,

no-button), state (move, pressed, released, drag) and coordinates. The dataset was used to evaluate a user verification system, as described in [17].

Our novel mouse dynamics dataset (ReMouse), which we are introducing in this paper and making available to the public, has been collected by means of a web platform developed using the Django REST framework. To collect mouse data from genuine human participants, the platform was deployed on MTurk (for more details, see Section 3.2).

The main differences between our ReMouse dataset and the mouse dynamics datasets previously released by other researchers are as follows: (i) The ReMouse dataset contains the mouse dynamics information of 100 users of mixed nationality, residing in diverse geographical regions, and using different devices (hardware and software components). (ii) The dataset contains dozens of 'repeat sessions' per each user, where 'repeat sessions' are sessions during which the user is asked to complete the same logical task in a guided online environment (e.g., play an online game involving the same sequence of steps and intermediate objectives). Through analysis of such 'repeat sessions', it is possible to obtain a better insight into the actual impact of 'repetition' on the user's mouse behavior (e.g., mouse trajectory and speed). According to our knowledge, this is the first dataset of this kind offered to the public. (iii) Each session in the ReMouse dataset is depicted with more granular information relative to the sessions in other datasets. Namely, in addition to the timing and positioning information of the mouse cursor, our dataset also contains mouse movement speed/velocity, the applications' window size (the height and width), as well as the anonymized IP addresses of the participants as user IDs.

Table 1 compares the characteristics of the most commonly studied publicly available dataset with those of our novel ReMouse dataset.

Table 1. The characteristics of the most prevalent publicly available dataset, including our novel ReMouse dataset.

Name	Ref.	# User	Data Collection	Period of Observing Each User's Activity	Action	Session Fields	Task	Repeat Sessions
Balabit	[28]	10	N/A	N/A	Mouse movement, point click, drag and drop	Timestamp, coordinates, pressed button, state of the mouse	Non-guided	No
Bogazici	[29]	24	1 month	2550 h	Mouse movement, point click, drag and drop	Timestamp, coordinates, button, state of the mouse, application window name	Non-guided	No
The Attentive Cursor	[30]	3K	N/A	2 h	Mouse movement, point click	Timestamp, coordinates, event name, XPath of the DOM element that relates to the event, the DOM element attributes (if any)	Guided	No
SapiMouse	[31]	120	N/A	4 min of each user's activity	Mouse movement, point click, drag and drop	Timestamp, coordinates, button, state of the mouse	Guided	No
Chao Shen	[32]	28	2 months	30 sessions of 30 min	Mouse movement, point click, drag and drop	Timestamp, action type, application type, screen area, window position	Non-guided	No
DFL	[17]	21	7 months	Daily users' mouse activities for 7 months	Mouse movement, point click, drag and drop	Timestamp, coordinates, button, state of the mouse	Non-guided	No
ReMouse	[9]	100	2 Days	5 min of each user's activity	Mouse movement, point click, drag and drop	User ID, session ID, timestamp, coordinates, button, event type, state of the mouse, speed, screen size	Guided	Yes

3. ReMouse Dataset

3.1. Web Platform for Data Collection

Our interactive web platform, which was developed for the purpose of mouse dynamics data collection, is hosted on AWS (Windows Server IIS) and is accessible through the following URL: http://human-likebots.com (accessed on 25 February 2023). On the front/user-facing end, the platform simulates a simple 'Catch Me If You Can!' online game (refer to Figure 1). The game web-page contains a JavaScript code which captures the actual mouse dynamics data (i.e., mouse move, load, click, scroll, ... events) as well as the associated metadata. Specifically, in the time interval during which the user stays on the web-site and plays the 'Catch Me If You Can!' game, the script preforms a discrete 'event polling' of various event listeners every 30 ms. In addition to recording the mouse-dynamics-related events, the script also captures the timestamps and x–y coordinates of the recorded events, mouse speed, session ID and screen size. The data collected by the script are first buffered and then sent to the back-end server every few seconds (we decided against shorter sampling and transmission intervals to avoid unnecessary data overhead). Using the Django Rest Framework [34], the server-side web application is able to receive and store the recorded event data in a log file (CSV format). The client- and server-side applications do not record any personal information about the users interacting with the human-likebots.com site.

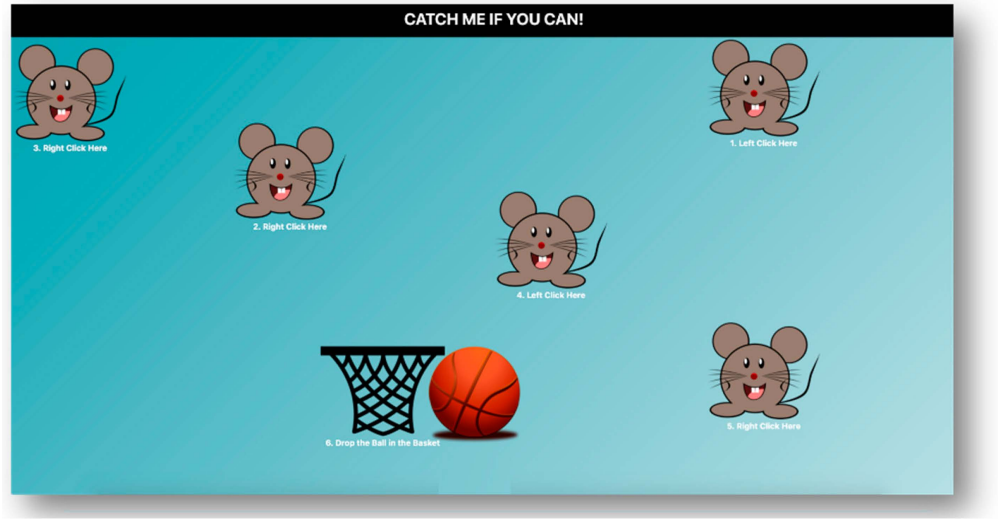


Figure 1. The web-site 'Catch Me if You Can!'.

3.2. ReMouse Dataset Acquisition

In order to collect real human-user data, our interactive human-likebots.com page was deployed on the Amazon MTurk platform (MTurk is a crowdsourcing marketplace that allows researchers to hire anonymous virtual workers to complete human intelligence tasks for pay. Currently, MTurk offers access to over 500,000 virtual workers from 190 countries). We specifically requested 100 MTurk users to visit and interact with our 'Catch Me If You Can!' site by playing multiple rounds of the game—for a total duration of 5 min. In each round of the game, the users were asked to follow six steps and perform three different actions, including left-click, right-click and drag-and-drop actions. We considered each round played by a particular user as a separate mouse movement session. Figure 2 shows the total number of sessions generated by each participating user, while Figure 3 shows the minimum, maximum and average session counts over all 100 users.

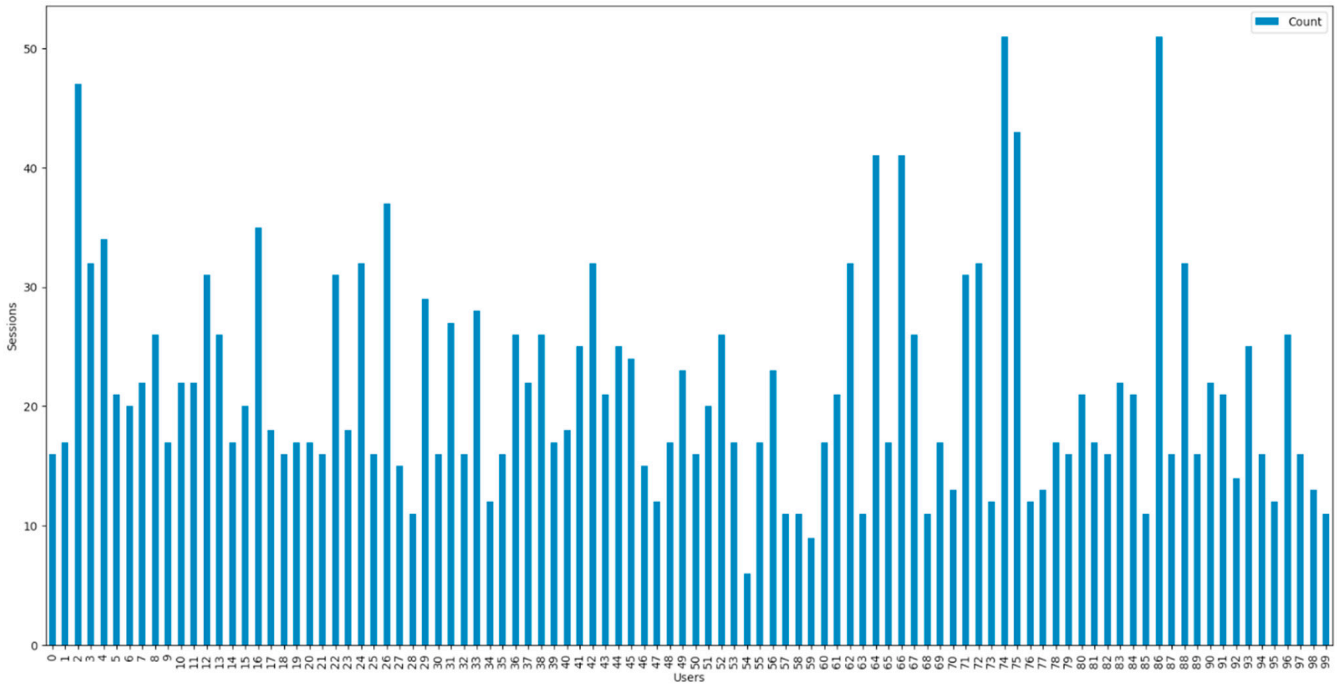


Figure 2. The number of sessions generated by each user.

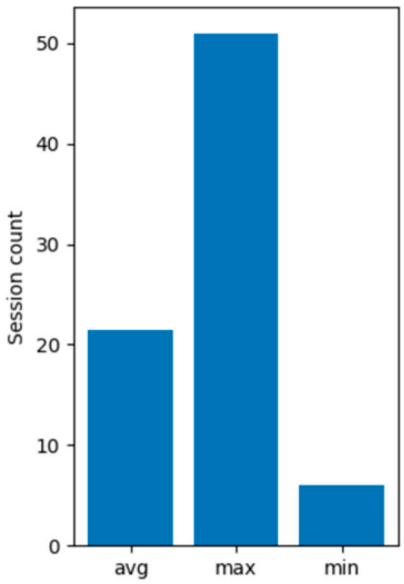


Figure 3. Session status.

4. ReMouse Dataset Analysis

4.1. Sessions Generated by The Same User

In the first stage of our ReMouse dataset study, we focused on analyzing the sessions generated by each individual user in isolation from other users. For the purpose of this analysis, a mouse cursor trajectory of a particular session was modeled by means of two time-dependent variables: (1) 2D coordinates/position of the mouse cursor; (2) speed of mouse cursor. As an illustration, Figure 4 displays the trajectories comprising only the mouse coordinates (i.e., positional information) of session number 3 for ReMouse users 90 to 98.

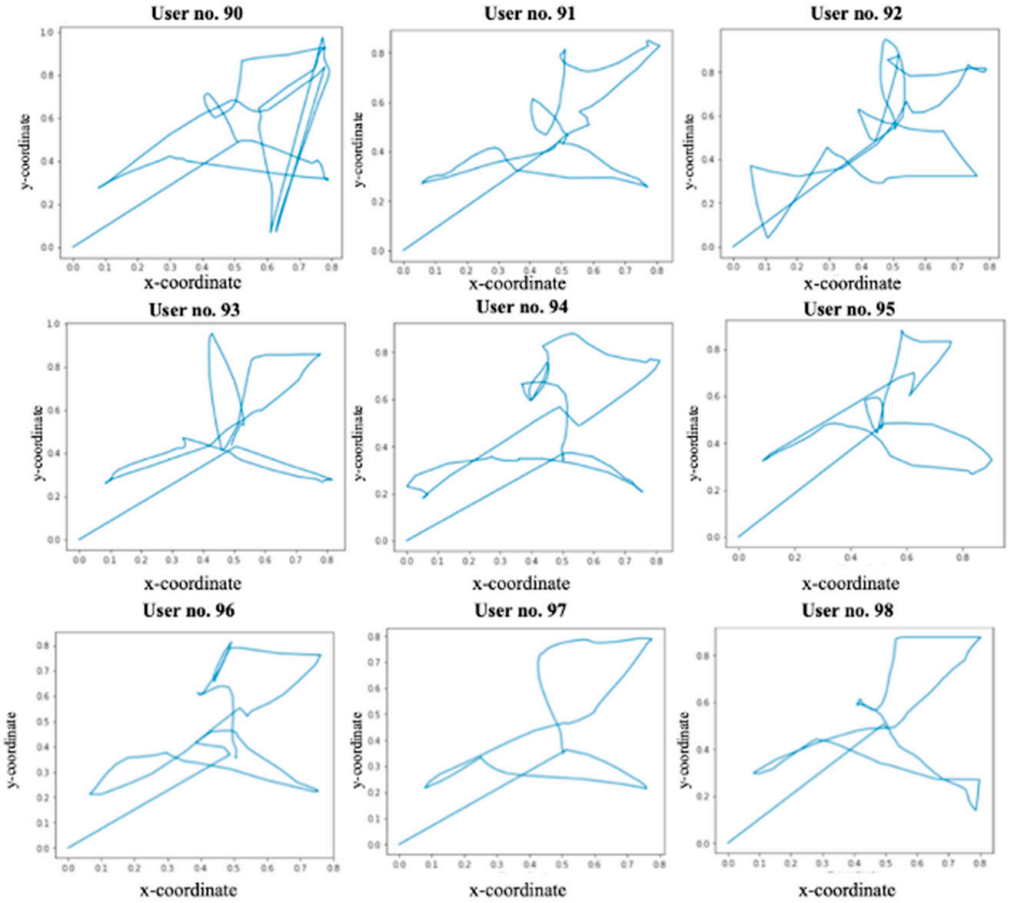


Figure 4. Visual representation of mouse cursor trajectory in the session with order number 3 for users 90 to 98.

Our analysis of single-user sessions led to some interesting observations:

Observation 1.1: It is evident from the collected data that by repeating the same online task over time (i.e., repeating multiple rounds of our 'Catch Me If You Can!' game), each user generally becomes faster and able to complete every subsequent round of the game in a progressively shorter amount of time. These findings are illustrated in Figure 5, which displays the 'time taken' and the 'average mouse movement speed' for user 82 (which is randomly chosen among the 100 participants) across each of the 16 rounds/sessions of the game that this particular user has performed. The same observation is also evident from Figure 6, which shows the dynamic time warping (DTW) distances [35] between the trajectories of subsequent pairs of sessions generated by user 82 (e.g., trajectories of first and second session, second and third session, etc.). As can be seen in Figure 6, the DTW distances between the trajectories of subsequent sessions become closer and shorter as the user keeps repeating the same task.

Note that we opted for the use of the DTW distance metric in our analysis as it has allowed us to measure the distance between two sessions (two time series) of different lengths and different time-wise alignments (DTW re-aligns two feature vector sequences by warping the time axis iteratively until an optimal match between the two sequences is found [35]). Figure 7 provides a closer look into the trajectories of two particular sessions (number 13 and 14) of user 82 and their respective DTW cumulative distance.

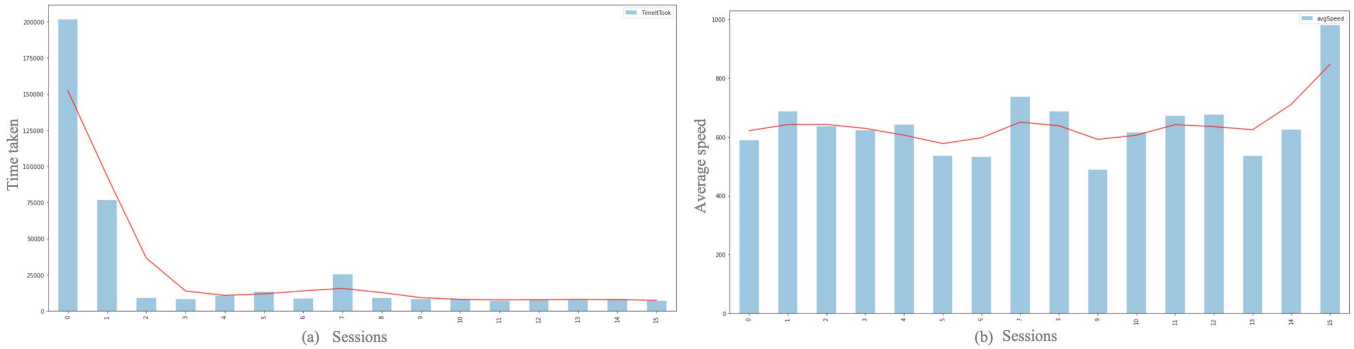


Figure 5. (**a**) Time taken to complete each of 16 conducted sessions for user number 82; (**b**) Average mouse movement speed for each of 16 conducted sessions.

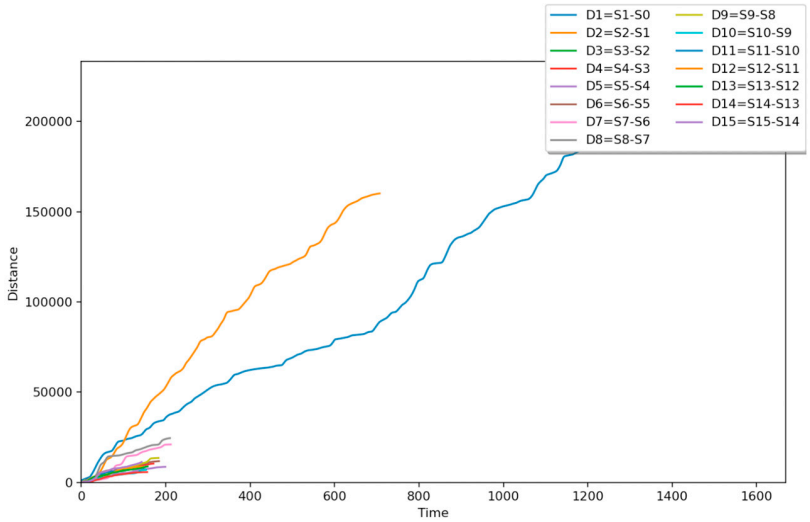
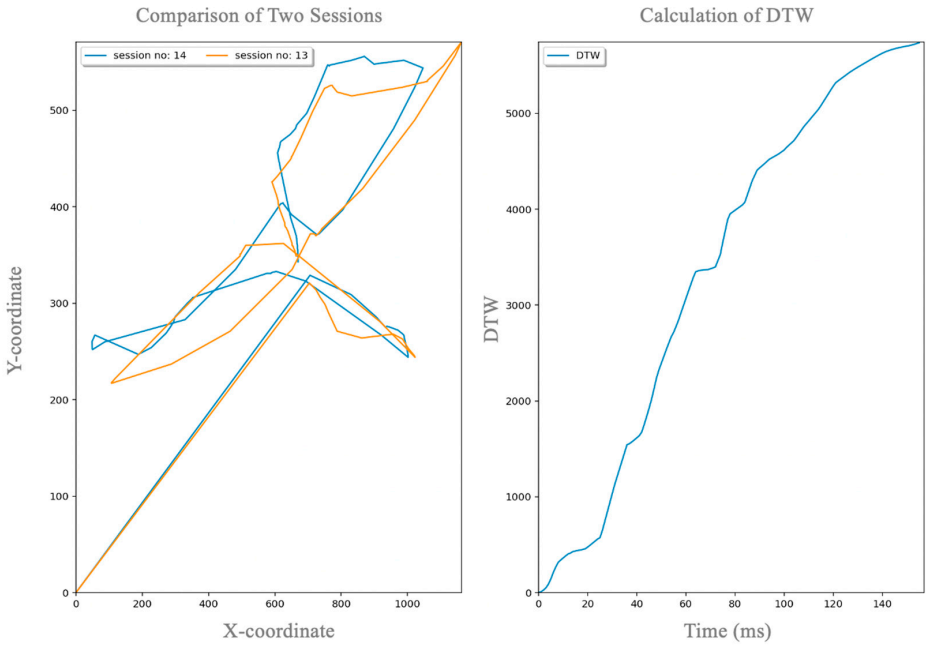


Figure 6. Cumulative difference/distance between subsequent pairs of sessions generated by user 82.



(**a**) (**b**)

Figure 7. (**a**) Trajectories of sessions 13 and 14 of user 82; (**b**) Cumulative DTW distance between two sessions.

To confirm Observation 1.1, we also deployed simple 'trend line analysis' [36] on the ReMouse dataset. A trend line is a bounding line that captures a trend and rallying patterns in a given dataset. If the slope of the line is a positive value, it indicates the trend is increasing, and a negative value implies that the trend is decreasing. We employed this analysis to discover the trend in 'time taken to complete a session' and 'average mouse speed' in relation to the session order number for each participating user. The average value of the slope in 'time taken to complete a session' trend lines, when calculated across all the users, was 417.0, which is a good indication that with every subsequent session/repetition the users generally spent less time completing the task. On the other hand, the average value of the slope in the 'speed of mouse movement' trend lines, when calculated across all users, was 10.0, which is further proof that users generally became faster in completing a similar online task with every subsequent session/repetition.

Observation 1.2: Even though the repeat sessions generated by each particular user became progressively 'closer' (as illustrated in Figure 6), no user is able to produce two entirely identical consecutive mouse trajectories when repeating the same online task. This observation is illustrated in Table 2, which shows the ids of the two closest consecutive sessions generated by each respective user in the ReMouse dataset when measured using the minimum normalized cumulative DTW distance. Moreover, since the overall cumulative DTW distances will be greater when the sessions are longer—cumulating over time—we normalized the DTW distance values by the time taken to complete each pair of sessions (i.e., the trajectory time-wise length). That way, the time component does not affect the results, and the minimum DTW distances show the actual trajectories' closeness. A closer inspection of the values in Table 2 reveals that user 74 produced the most similar consecutive trajectories in the ReMouse dataset (corresponding to sessions number 39 and 40), with a normalized cumulative DTW distance of 64.23521268 (note that two identical sessions would produce a DTW distance of 0). The graph shown in Figure 8 plots the minimum normalized cumulative DTW distance values from Table 2, confirming Observation 1.2. Figure 9 provides a closer look at the trajectories of sessions 39 and 40 of user 74, as well as their respective normalized cumulative DTW.

Observation 1.3: Through the analysis of ReMouse dataset, we further observed that in the initial sessions the users acted generally more confused, i.e., their cursors exhibited more 'erratic' behavior until the users finally figured out what exactly they were expected to do. However, even in these initial sessions, the mouse speed was not considerably slower than in the later session, which is indicated through a relatively small positive slope value obtained from the 'trend line analysis'.

Table 2. The most similar trajectories generated by each participating user in the ReMouse dataset with their respective DTW values—the minimum DTW normalized cumulative distance between the closest sessions.

Users	Sessions	Min DTW Normalized Cumulative Distance	Users	Sessions	Min DTW Normalized Cumulative Distance
0	7,8	591.6516	50	2,3	303.9826
1	5,6	295.2985	51	4,5	291.6989
2	35,36	147.0755	52	7,8	272.5094
3	13,14	192.1207	53	13,14	196.9675
4	9,10	180.0245	54	2,3	1490.494
5	4,5	398.1191	55	13,14	421.657
6	8,9	272.4871	56	11,12	276.5871
7	19,20	293.7516	57	8,9	1387.489
8	17,18	192.9701	58	8,9	634.1661
9	11,12	345.1108	59	6,7	777.4243
10	5,6	308.2797	60	6,7	174.8066

Table 2. *Cont.*

Users	Sessions	Min DTW Normalized Cumulative Distance	Users	Sessions	Min DTW Normalized Cumulative Distance
11	3,4	572.3161	61	17,18	232.3106
12	2,3	107.556	62	27,28	126.1892
13	21,22	262.7717	63	3,4	1112.61
14	4,5	297.0564	64	33,34	142.0399
15	2,3	287.2074	65	9,10	301.4555
16	9,10	116.766	66	33,34	199.8493
17	10,11	247.4575	67	14,15	137.9862
18	12,13	275.4263	68	3,4	1728.454
19	9,10	371.7259	69	4,5	427.3393
20	7,8	175.7365	70	9,10	1201.285
21	11,12	280.7912	71	17,18	126.8211
22	23,24	127.987	72	16,17	211.9789
23	7,8	343.7548	73	5,6	487.4164
24	28,29	198.9364	**74**	**39,40**	**64.23521**
25	12,13	358.7146	75	24,25	85.11796
26	29,30	204.9529	76	8,9	402.6993
27	11,12	241.8954	77	3,4	623.3006
28	7,8	462.876	78	10,11	412.5679
29	26,27	110.2986	79	11,12	355.0567
30	5,6	210.5634	80	18,19	488.2605
31	11,12	203.5428	81	7,8	315.7737
32	5,6	213.7062	82	13,14	383.0098
33	14,15	258.7817	83	9,10	262.1923
34	8,9	503.8331	84	6,7	275.4376
35	2,3	241.2987	85	8,9	2391.673
36	23,24	210.416	86	48,49	174.3101
37	10,11	305.7957	87	11,12	422.6979
38	23,24	112.3997	88	24,25	113.6169
39	4,5	191.0098	89	7,8	354.2762
40	7,8	429.8543	90	17,18	134.8357
41	17,18	143.9127	91	6,7	299.5449
42	21,22	318.2114	92	5,6	792.4915
43	18,19	226.5839	93	7,8	292.0623
44	4,5	446.748	94	8,9	282.6595
45	6,7	181.1306	95	9,10	432.2253
46	6,7	240.4841	96	23,24	210.416
47	5,6	630.878	97	13,14	261.8753
48	12,13	294.704	98	2,3	753.1881
49	2,3	315.2712	99	8,9	386.572

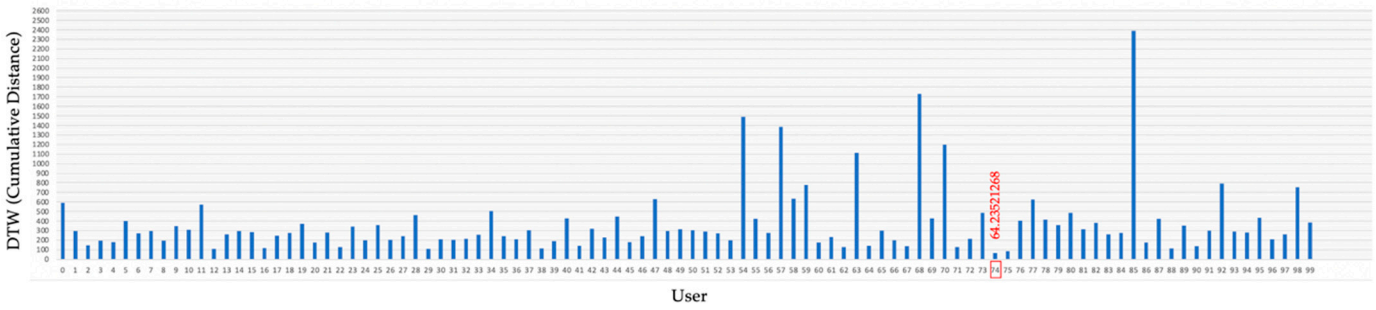
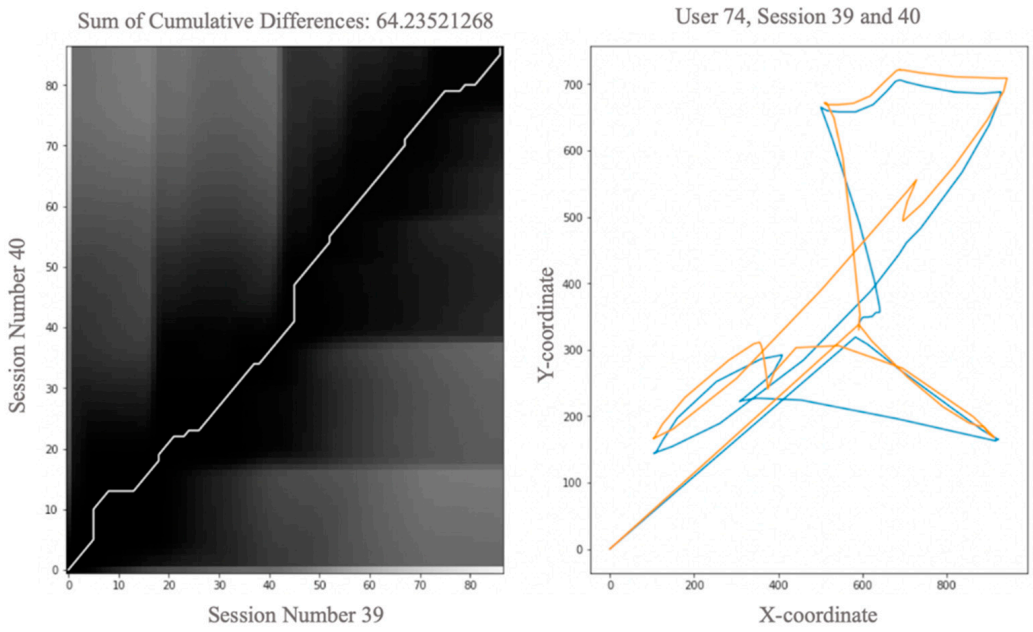


Figure 8. Minimum DTW normalized cumulative distances across sessions of each individual user.



(**a**) (**b**)

Figure 9. (**a**) Sum of cumulative DTW distance value in sessions generated by the same user, user 74; (**b**) Sessions 39 (blue) and 40 (orange) of user 74.

4.2. Sessions Generated by Different User

In the second stage of our ReMouse dataset study, the focus was on the pairwise analysis of sessions generated by different users. The findings of this analysis are summarized below:

Observation 2.1: Different users produced different-looking sessions when completing the same/similar online task.

The validity of this observation was confirmed by comparing all users' sessions in our dataset (i.e., by calculating the cross-user pairwise minimum DTW distance). Table 3 shows the minimum normalized cumulative DTW distance value between two sessions of two distinct users out of all users' sessions. As shown, the most similar trajectories generated by two distinct users are sessions 6 and 29 of users 1 and 2, respectively. The actual DTW distance between these sessions is 21.94, which suggests that, although similar, these two sessions are not identical. This observation can be further generalized, implying that even though sessions generated by two distinct human users while completing the same/similar online task may exhibit a high degree of similarity, they are also likely to be sufficiently distinct from each other.

Table 3. Cross-user pairwise DTW normalized cumulative distance calculation result.

Min DTW	Users	Sessions
21.941833	1 and 2	6 and 29

Observation 2.2: There are no two sessions created by two distinct users that are closer to each other than (any) two sessions created by the same user when completing the same/similar online task.

To confirm this observation, in addition to calculating the distance between sessions generated by different users, we also computed the minimum normalized cumulative DTW distance between ANY two (not just consecutive) sessions generated by the same user in the ReMouse dataset. Table 4 summarizes these results, and it shows that out of the entire ReMouse dataset, user 1 has generated two most similar trajectories (corresponding to sessions number 16 and 28) with a respective distance of 20.376812.

Table 4. Pairwise DTW normalized cumulative distance calculation result—the same user.

Min DTW	Users	Sessions
20.376812	1 and 1	16 and 28

The observations of this section can be further generalized and put in the context of session-replay bots. Namely, the numerical results obtained through the analysis of ReMouse dataset imply that no two sessions (i.e., mouse trajectories) generated on a static web-site—regardless of whether they are generated by the same or two distinct users—can be identical. Based on this, we further hypothesize that only pre-programmed session-replay bots are theoretically able to produce identical browsing sessions (i.e., mouse trajectories). Or, put another way, any occurrence/observation of 'identical' or 'almost identical' browsing sessions (i.e., mouse trajectories) in a web-site should be taken with caution, potentially warranting further investigation for the presence of session-replay bots.

5. Feature Engineering—Preparing ReMouse Dataset for Machine-Learning-Based Analysis

In previous studies on mouse dynamics, researchers have commonly relied on heuristics-based (i.e., manually selected) mouse movement features, such as 2D cursor position, mouse speed, click frequency, etc. The results of our own ReMouse dataset analysis using manually selected features are presented in Section 4. However, some known challenges of manual features selection are: (1) manual feature selection requires in-depth expert knowledge of the specific dataset at hand and the ultimate application environment; (2) there is often a need to fine-tune the number and type of manually selected features for each dataset, which tends to be a time-consuming process; (3) the generalization value of the results obtained using manual feature selection is often questionable. One of the objectives of our work was to analyze the ReMouse dataset by means of advanced machine learning (ML) techniques. However, for the reasons outlined above, we were determined to avoid basing our ML analysis on manually selected features. Additionally, due to the different durations of individual user sessions in the ReMouse dataset, we were facing very heterogeneous 'mouse location' and 'mouse speed' feature vector representations (i.e., the feature vectors representing different sessions were of variable/non-fixed length). Training an ML algorithm using such non-uniform set of feature vectors would have required additional expert-knowledge decision making and the manual re-engineering of input data.

As an alternative to manual feature selection and feature vector re-engineering, and inspired by works [2,22], we pursued a novel approach to representing individual user sessions in the ReMouse dataset. Namely, in this part of our analysis, rather than manually extracting features to describe a user's unique mouse behavior characteristics, we mapped the mouse trajectories into pictures. In order to conduct automated feature extraction on image representations of user sessions from the ReMouse dataset, we deployed a pre-trained deep learning model—VGG16 [37]. In particular, we used the VGG16 library implemented in Keras [38]. VGG16 is a convolutional neural network model well known for its ability to perform very-high-accuracy feature extraction on image datasets [39]. The reason why we resorted to deploying a pre-trained VGG16 model is the fact that working with a 'from-scratch' convolutional neural network may require days of training and millions of images to achieve a high accuracy in real-world applications [40] (from the perspective of image processing, our ReMouse dataset is of relatively small size, containing the sessions of 'only' 100 users). For the purposes of our research, we acquired the generic pre-trained VGG16 model from [38] and retrained it on our own image representations of web sessions from the ReMouse dataset (the process of re-using the weights from a pre-trained model is called 'Transfer Learning' [41]). The original VGG16 model used in our work was trained on standard computer vision benchmark datasets, including ImageNet [42].

Using VGG16, we ended up with each image (i.e., user session) being represented as a vector with 1000 features [43]. To further reduce the number of features identified with VGG16, next, we used principal component analysis (PCA) [44]. PCA produced 100 eigenvectors over the VGG16 feature space. Nevertheless, as shown in Figure 10, not all of the 100 identified PCA eigenvectors are of the same significance, as 95% of data variance occurs over the first 57 eigenvectors. Thus, for the purpose of our ML-based analysis (as discussed in the next section) we opted to map our original ReMouse dataset into a set of feature vectors over the first 57 most significant PCA eigenvectors.

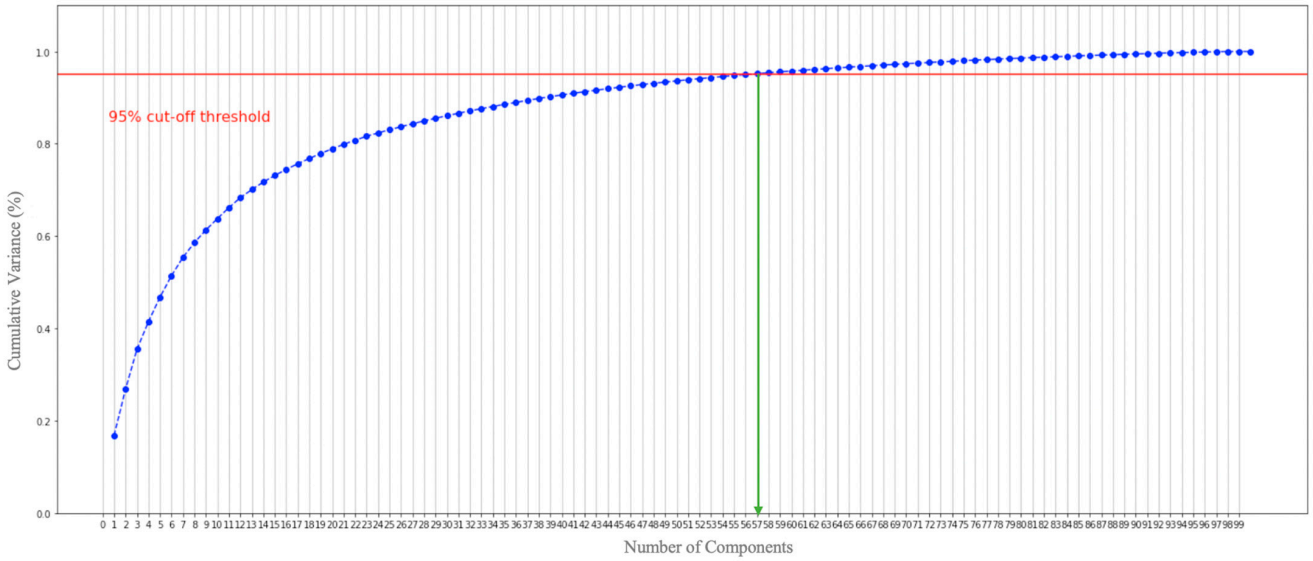


Figure 10. The number of components needed to explain the variance.

6. ML-Based Analysis of ReMouse Dataset: Focusing on Sessions Generated by Different Users

The objective of our ML-based analysis of the curated image-based ReMouse dataset (as explained in Section 5) was to investigate the (dis)similarities between comparable (same-order number) web sessions generated by different users. We specifically decided to look at the third session generated by each of the 100 participating ReMouse users (forming one data subset, which we will refer to as 'ReMouse Subset-3' in the reminder of this article), as well as the fifth session generated by each of the 100 participating ReMouse users (forming the second data subset, which we will refer to as 'ReMouse Subset-5'). We opted to look at the third and fifth sessions due to our observation that for most ReMouse users some of the originally exhibited 'erratic' mouse behavior largely disappears after the first two rounds/repetitions of the 'Catch Me If You Can!' game (see Section 3). In other words, the user behavior and mouse trajectory in these sessions are generally 'stable' and thus likely to produce more accurate results. To conduct the cross-user session (dis)similarity analysis, we specifically decided to deploy unsupervised ML learning, including the Self-Organizing Map (SOM) and several unsupervised clustering ML algorithms.

The SOM algorithm is typically used to build a topology-preserving mapping of high-dimensional input data to 2D or 3D space, where the similarity of individual input points can be assessed in more intuitive (visual and non-visual) ways. Unsupervised clustering is known for its ability to decompose a dataset into subgroups based on their similarity so that data points in the same cluster are more closely related to each other than data points in different clusters [45].

According to our knowledge, this is the first research study that has looked into the use of unsupervised clustering on the image representation of user sessions for the purpose of cross-user session (dis)similarity analysis. Additionally, the only other work that has pursued image-based web-session representation and analysis [22] was specifically concerned with the problem of malicious web-bot detection through session classification,

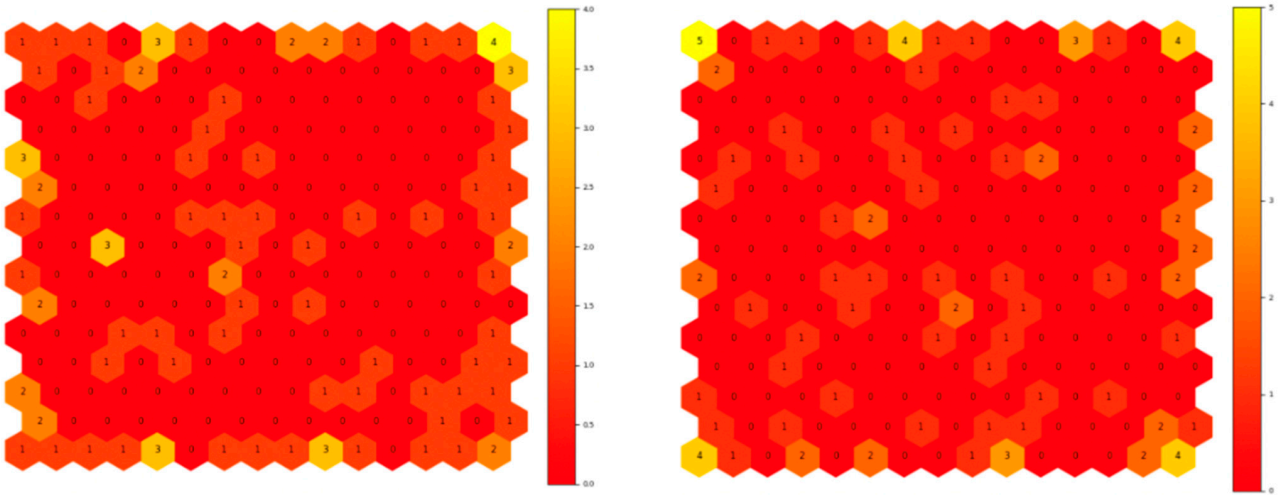
and thus ultimately opted for the use of supervised deep learning—as opposed to the question of session similarity, which is the focus of our work and requires the use of unsupervised techniques.

6.1. Data Analysis Using SOM Map

The Self-Organizing Map (SOM) algorithm [46] is generally used to create a 2D topology-preserving and density-mapping representation of a multi-dimensional input (i.e., training) dataset. The topology preservation property implies that if two input points end up firing nearby nodes in the trained SOM map during the deployment phase then the two points are relatively close to each other (i.e., are similar) in the original input space. On the other hand, the density-mapping property means that the regions of high-input-dataset density are mapped to SOM regions with more neurons.

For the purposes of our research, we trained two 15-by-15-sized SOM maps (experimentally), one using the ReMouse Subset-3 and the other using ReMouse Subset-5. We used the SOM implementation from the Python SOMPY package [47], which has a structure similar to *somtoolbox* in MATLAB. In terms of functionalities, the package uses only batch training (which is faster than online training) and *sklearn* or random initialization.

The heatmaps generated on each of the two trained SOM maps are shown in Figures 11a and 11b, respectively. An SOM heatmap is produced by displaying how many of the training inputs are associated with each node in the trained SOM map [48]. It is very evident from the two heatmaps that there are no actual (i.e., distinguishable) clusters in either ReMouse Subset-3 or ReMouse Subset-5—as most neurons are 'fired' by no/one single-input point, and only a handful of neurons are fired by two or more (distinct) input points. It should also be noted that the neurons with an input-data membership of two or more are largely distributed at the edges of the respective SOM maps, which suggests that the actual 'closeness' of the input points that fire these neurons may not be significant. Border neurons in an SOM map do not 'stretch out' during the training process as much as they should, and as a result they tend to 'attract' many potentially very different/distant points located on the 'outside' of the SOM border. This phenomenon in known in the literature as the 'SOM border effect' [49].



(**a**) (**b**)

Figure 11. Users' data points map: (**a**) session number 3; (**b**) session number 5.

From a practical point of view, that such a disperse distribution of data points form ReMouse Subset-3 and ReMouse Subset-5 (as shown in Figure 11a,b) is a clear indication that individual users—when performing the same general online task—are likely to end up producing very different/distinct mouse trajectories. When put in the context of session-replay bots, this further suggests that any session/trajectory that shows a significant

similarity with an already-observed session/trajectory should be flagged as potentially 'malicious', since (according to our results) the likelihood that both of such sessions are genuinely human is rather small.

As part of our future work, we plan to deploy different variants of the SOM algorithm (e.g., growing SOM map [50] and evolving SOM algorithm [51]) in order to further address the issue of the 'border effect' observed in our dataset.

6.2. Data Analysis Using Unsupervised Clustering Techniques

In order to validate our initial findings obtained by means of SOM heatmaps, we further performed an unsupervised clustering of ReMouse Subset-3 and ReMouse Subset-5 using the SOM clustering [47] (the python package provides an additional feature which enables automated identification of the main clusters within the formed map using K-means clustering algorithm), K-means clustering [52], and agglomerative clustering [53] algorithms.

An important result coming out of this stage of our research is obtaining the Silhouette and Davies–Bouldin scores, which were obtained by performing clustering on the two data subsets with a gradually increasing number of assumed clusters [54,55]. The Silhouette score measures how similar an object is to its own cluster (cohesion) compared with other clusters (separation). A higher Silhouette value implies that points are well matched to their own cluster and poorly matched to neighboring clusters. The Davies–Bouldin score is the average similarity measure of each cluster with its most similar cluster. Clusters that are farther apart and less dispersed will result in a higher Davies–Bouldin score.

Figures 12 and 13 depict the Silhouette and Davies–Bouldin score obtained using K-means clustering algorithms. Similar results have been obtained with the other two clustering algorithms. In the cases of all three algorithms, the highest values of the two scores are recorded for k = 2, suggesting that the optimal number of clusters is two. Figures 14–16 provide 2D and 3D visualizations of the actual clustering results obtained on ReMouse Subset-3 and ReMouse Subset-5 using the three selected clustering algorithms and assuming k = 2. All three figures provide clear evidence that, even under the optimal number of clusters (k = 2), the input data is pretty spread out throughout the input space, and many points that formally belonging to the same cluster are at a significant distance from each other. This further supports our earlier hypothesis that session trajectories generated by different users while completing the same online task are sufficiently distinguishable from each other.

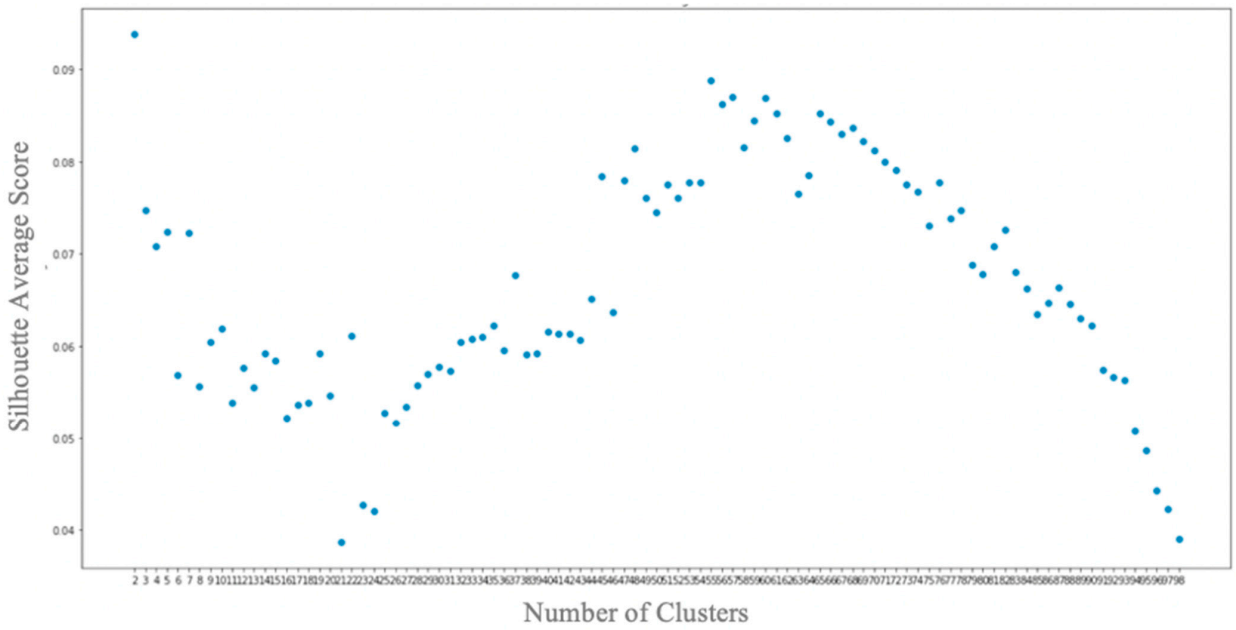


Figure 12. Silhouette average score.

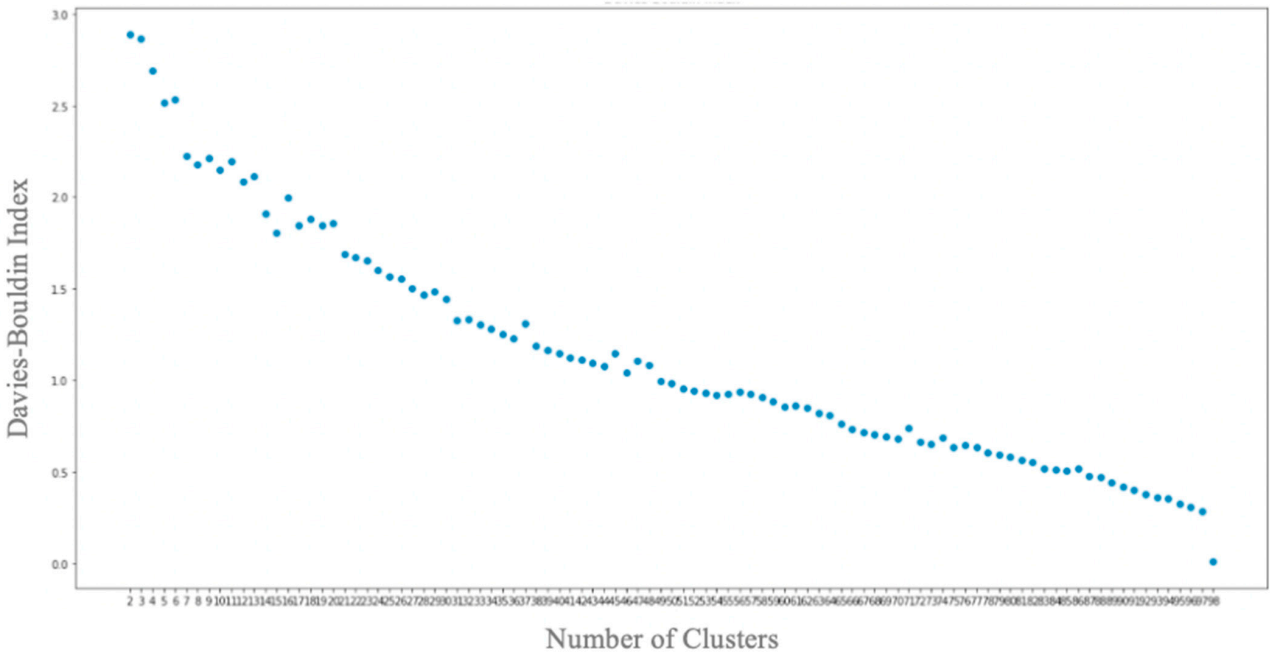


Figure 13. Davies–Bouldin index.

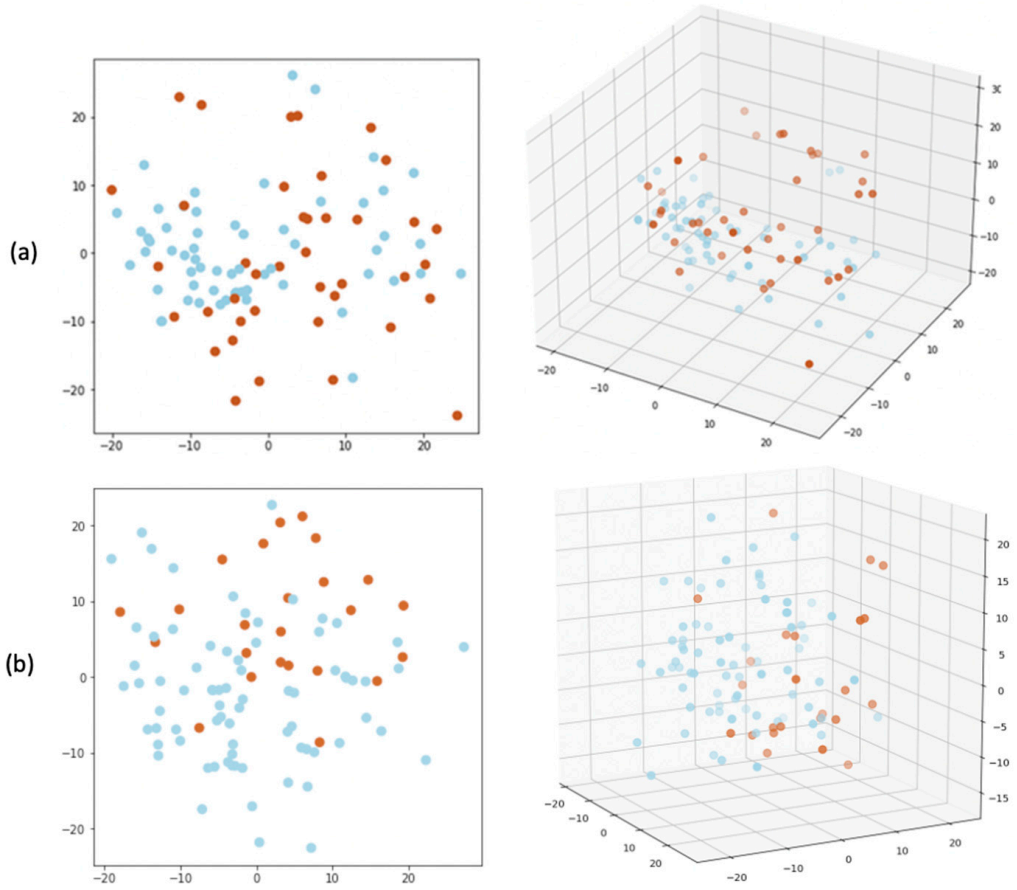


Figure 14. Unsupervised clustering visualization using SOM: (**a**) session number 3 and (**b**) session number 5 of all users.

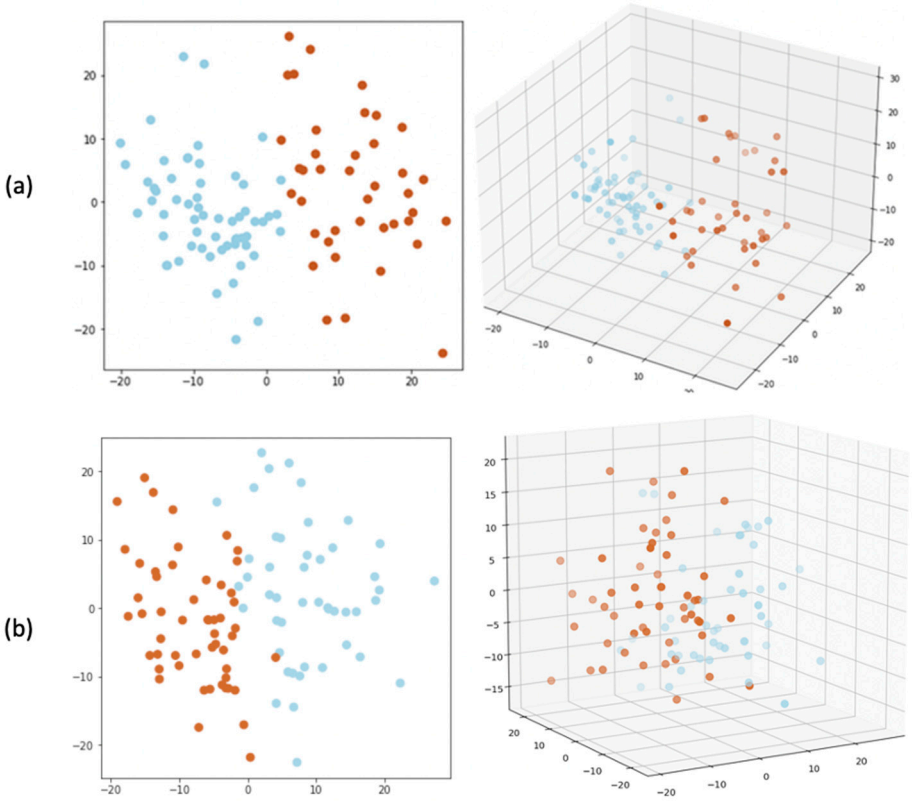


Figure 15. Unsupervised clustering visualization using K-means clustering algorithm, (**a**) session number 3 and (**b**) session number 5 of all users.

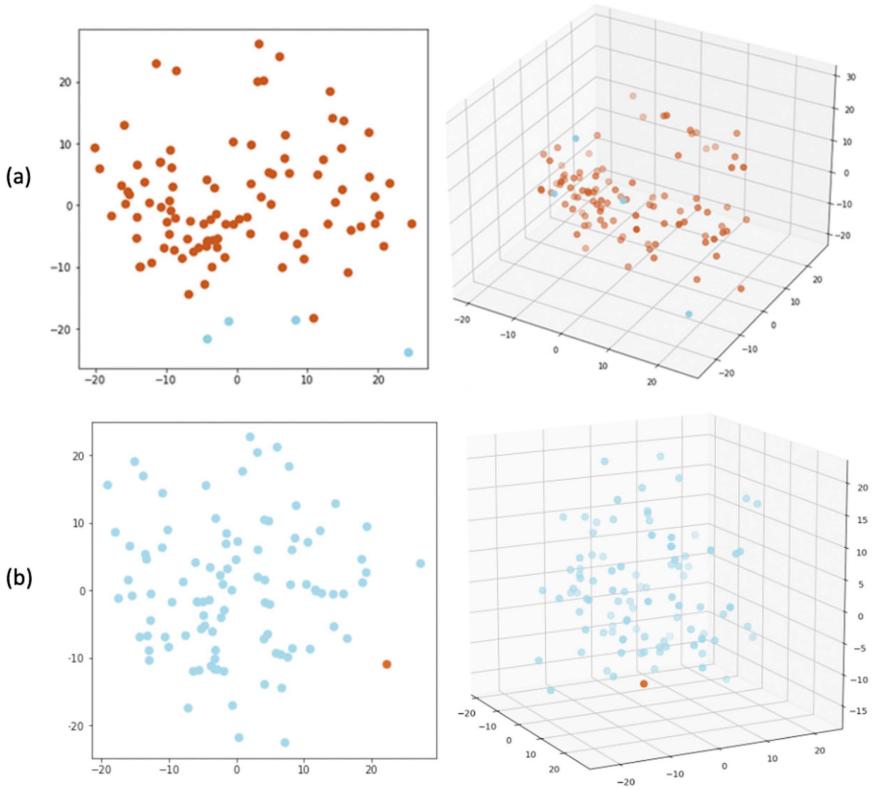


Figure 16. Unsupervised clustering visualization using agglomerative clustering algorithm, (**a**) session number 3 and (**b**) session number 5 of all users.

7. Conclusions and Future Work

In this work, we presented an in-depth analysis of our novel real-world mouse dynamics dataset, the ReMouse dataset. We began by reviewing the literature that investigated mouse dynamics in the context of user authentication and bot detection. We also provided a summary of several publicly available mouse dynamics datasets. We then analyzed the ReMouse dataset using statistical and advanced ML-based methods, including deep and unsupervised neural learning.

In the first stage of the preliminary analysis using statistical methods, we focused on analyzing the sessions generated by each individual user in isolation from other users. Second, the focus was on the pairwise analysis of sessions generated by different users. Based on the preliminary analysis of our novel ReMouse dataset, we concluded that although sessions generated by genuine human users are relatively similar to each other, there always exist some minimum distinguishable differences between them. We showed that sessions whose 'difference' from each other is below the determined threshold should potentially be flagged as 'replay' sessions generated by session-replay bots.

Considering the fact that the generalization value of the results obtained using manual feature selection is often questionable, we then investigated the (dis)similarities between comparable (same-order number) web sessions generated by different users by means of advanced machine learning techniques. The results further supported our earlier hypothesis that session trajectories generated by different users while completing the same online task are sufficiently distinguishable from each other.

According to our knowledge, the ReMouse dataset is the first publicly available mouse dynamics dataset containing repeat sessions generated by the same human user(s). As such, this dataset can be a very valuable resource for research studies that aim to improve our understanding of (human) user behavior during repetitive interactions with the same web-site, with the ultimate goal of developing effective techniques for the detection of, and defense against, sessions-replay bots.

We believe that the ReMouse dataset contains enough statistical data to facilitate unbiased and high-quality research in the above-mentioned research areas. However, we also would like to point out a few possible, though minor, limitations of our dataset and work. One potential limitation of our dataset/work can be related to the platform we used to collect the data, MTurk. Although MTurk workers are generally pretty diverse when it comes to their place of residence or profession, they tend to be less diverse in terms of their age, education, computer-use proficiency, etc. [56]. This can complicate how data can be interpreted, affecting the reliable and validity of our conclusions, as well as the generalizability of such results.

Nevertheless, more importantly, this study is the first of its kind, so it effectively demonstrates the importance of filling the literature gaps, highlighting the need for further development in the area of our study. This work aims to bring more attention to the problems/threats posed by session-replay web bots, which carry out the most advanced types of malicious web bot attacks. Therefore, we invite other researchers to work alongside us. We made some progress in providing the data and tools and hope to facilitate further studies by other researchers.

For future work, we plan to extend our image-based ML analysis of the ReMouse dataset by considering other aspects of mouse dynamics rather than just trajectory (e.g., by additionally embedding the information on time, mouse velocity and click events into the image representation of a user session). We are also currently working on incorporating the malicious sessions generated by actual session-replay bots into the ReMouse dataset. Finally, we plan to experiment with different variants of the SOM algorithm (e.g., growing an SOM map and evolving the SOM algorithm) in order to further address the issue of the 'border effect', which has been observed in our preliminary analysis.

Author Contributions: Conceptualization, S.S. and N.V.; methodology, S.S.; validation, S.S. and N.V.; writing—original draft preparation, S.S.; writing—review and editing, S.S. and N.V.; supervision, N.V.; project administration, N.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board (OFFICE OF RESEARCH ETHICS (ORE)) of York University (certificate #: e2022-374 issued on 4 August 2022).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Our novel ReMouse dataset presented in this study is openly available in the IEEE Dataport at https://dx.doi.org/10.21227/jkmt-za31, accessed on 25 February 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Maureen. What Is Behavioral Biometric Authentication? 1Kosmos. 2022. Available online: https://www.1kosmos.com/biometric-authentication/what-is-behavioral-biometrics-authentication/ (accessed on 25 February 2023).
2. Thomas, P.A.; Mathew, K.P. A Broad Review on Non-Intrusive Active User Authentication in Biometrics. *J. Ambient. Intell. Human Comput.* **2023**, *14*, 339–360. [CrossRef] [PubMed]
3. Leiva, L.A.; Arapakis, I.; Iordanou, C. My Mouse, My Rules: Privacy Issues of Behavioral User Profiling via Mouse Tracking. In Proceedings of the 2021 Conference on Human Information Interaction and Retrieval, 51–61. CHIIR '21, Canberra, ACT, Australia, 14–19 March 2021; Association for Computing Machiner: New York, NY, USA, 2021. [CrossRef]
4. Kaklauskas, A. Web-based Biometric Computer Mouse Advisory System to Analyze a User's Emotions and Work Productivity. In *Biometric and Intelligent Decision Making Support*; Kaklauskas, A., Ed.; Intelligent Systems Reference Library; Springer International Publishing: Cham, Switzerland, 2014; Volume 81, pp. 137–173. [CrossRef]
5. Katerina, T.; Nicolaos, P. Mouse behavioral patterns and keystroke dynamics in End-User Development: What can they tell us about users' behavioral attributes? *Comput. Hum. Behav.* **2018**, *83*, 288–305. [CrossRef]
6. Rahman, R.U.; Tomar, D.S. Threats of price scraping on e-commerce websites: Attack model and its detection using neural network. *J. Comput. Virol. Hacking Tech.* **2020**, *17*, 75–89. [CrossRef]
7. Nick, R. How Attackers Use Request Bots to Bypass Your Bot Mitigation Solution. Security Boulevard (Blog). 2021. Available online: https://securityboulevard.com/2021/07/how-attackers-use-request-bots-to-bypass-your-bot-mitigation-solution/ (accessed on 14 June 2022).
8. Chu, Z.; Gianvecchio, S.; Wang, H. Bot or Human? A Behavior-Based Online Bot Detection System. In *From Database to Cyber Security: Essays Dedicated to Sushil Jajodia on the Occasion of His 70th Birthday*; Pierangela, S., Indrajit, R., Indrakshi, R., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; pp. 432–449. [CrossRef]
9. Sadeghpour, S.; Vlajic, N. *ReMouse-Mouse Dynamic Dataset*; IEEE: New York, NY, USA, 2022; Available online: https://ieee-dataport.org/documents/remouse-mouse-dynamic-dataset (accessed on 24 August 2022).
10. Jaiswal, A.K.; Tiwari, P.; Hossain, M.S. Predicting users' behavior using mouse movement information: An information foraging theory perspective. *Neural Comput. Appl.* **2020**, 1–14. [CrossRef]
11. Kirsh, I.; Joy, M. Exploring Pointer Assisted Reading (PAR): Using Mouse Movements to Analyze Web Users' Reading Behaviors and Patterns. In *HCI International 2020-Late Breaking Papers: Multimodality and Intelligence*; Constantine, S., Masaaki, K., Helmut, D., Lauren, R.-J., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2020; pp. 156–173. [CrossRef]
12. Chong, P.; Elovici, Y.; Binder, A. User Authentication Based on Mouse Dynamics Using Deep Neural Networks: A Comprehensive Study. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 1086–1101. [CrossRef]
13. Hu, T.; Niu, W.; Zhang, X.; Liu, X.; Lu, J.; Liu, Y. An Insider Threat Detection Approach Based on Mouse Dynamics and Deep Learning. *Secur. Commun. Netw.* **2019**, *2019*, 1–12. [CrossRef]
14. Kaixin, W.; Liu, H.; Wang, B.; Hu, S.; Song, J. A User Authentication and Identification Model Based on Mouse Dynamics. In Proceedings of the 6th International Conference on Information Engineering, online, 19–20 November 2022; 2017; pp. 1–6.
15. Yildirim, M.; Anarim, E. Novel Feature Extraction Methods for Authentication via Mouse Dynamics with Semi-Supervised Learning. In Proceedings of the 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), Izmir, Turkey, 31 October–2 November 2019; 2020; pp. 1–6. [CrossRef]
16. Antal, M.; Fejer, N.; Buza, K. SapiMouse: Mouse Dynamics-based User Authentication Using Deep Feature Learning. In Proceedings of the 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 19–21 May 2021; pp. 61–66. [CrossRef]

17. Antal, M.; Denes-Fazakas, L. User Verification Based on Mouse Dynamics: A Comparison of Public Data Sets. In Proceedings of the 2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 23–31 May 2019; pp. 143–148. [CrossRef]
18. Antal, M.; Fejér, N. Mouse dynamics based user recognition using deep learning. *Acta Univ. Sapientiae Inform.* **2020**, *12*, 39–50. [CrossRef]
19. Yildirim, M.; Anarim, E. Mitigating insider threat by profiling users based on mouse usage pattern: Ensemble learning and frequency domain analysis. *Int. J. Inf. Secur.* **2021**, *21*, 239–251. [CrossRef]
20. Acien, A.; Morales, A.; Fierrez, J.; Vera-Rodriguez, R. BeCAPTCHA-Mouse: Synthetic mouse trajectories and improved bot detection. *Pattern Recognit.* **2022**, *127*, 108643. [CrossRef]
21. Iliou, C.; Kostoulas, T.; Tsikrika, T.; Katos, V.; Vrochidis, S.; Kompatsiaris, I. Detection of Advanced Web Bots by Combining Web Logs with Mouse Behavioural Biometrics. *Digit. Threat. Res. Pract.* **2021**, *2*, 1–26. [CrossRef]
22. Wei, A.; Zhao, Y.; Cai, Z. A Deep Learning Approach to Web Bot Detection Using Mouse Behavioral Biometrics. In *Biometric Recognition*; Zhenan, S., Ran, H., Jianjiang, F., Shiguang, S., Zhenhua, G., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2019; pp. 388–395. [CrossRef]
23. Rahman, R.U.; Tomar, D.S. New biostatistics features for detecting web bot activity on web applications. *Comput. Secur.* **2020**, *97*, 102001. [CrossRef]
24. Chuda, D.; Peter, K.; Jozef, T. Mouse Clicks Can Recognize Web Page Visitors! In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 21–22.
25. Niu, H.; Chen, J.; Zhang, Z.; Cai, Z. Mouse Dynamics Based Bot Detection Using Sequence Learning. In *Biometric Recognition*; Jianjiang, F., Junping, Z., Manhua, L., Yuchun, F., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2021; pp. 49–56. [CrossRef]
26. Rahman, R.U.; Tomar, D.S. A new web forensic framework for bot crime investigation. *Forensic Sci. Int. Digit. Investig.* **2020**, *33*, 300943. [CrossRef]
27. Solano, J.; Lopez, C.; Esteban, R.; Alejandra, C.; Lizzy, T.; Martin, O. SCRAP: Synthetically Composed Replay Attacks vs. Adversarial Machine Learning Attacks against Mouse-Based Biometric Authentication. In Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security, Virtual Event, USA, 13 November 2020; pp. 37–47.
28. Fülöp, Á.; Kovács, L.; Kurics, T.; Windhager-Pokol, E. Balabit Mouse Dynamics Challenge Data Set. 2016. Available online: https://github.com/balabit/Mouse-Dynamics-Challenge (accessed on 14 June 2022).
29. Kılıç, A.A.; Yıldırım, M.; Anarım, E. Bogazici mouse dynamics dataset. *Data Brief* **2021**, *36*, 107094. [CrossRef] [PubMed]
30. Leiva, L.A.; Arapakis, I. The Attentive Cursor Dataset. *Front. Hum. Neurosci.* **2020**, *14*, 565664. [CrossRef]
31. Antal, M. Sapimouse. Python. 2021. Available online: https://github.com/margitantal68/sapimouse (accessed on 14 June 2022).
32. Shen, C.; Cai, Z.; Guan, X. Continuous authentication for mouse dynamics: A pattern-growth approach. In Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012), Boston, MA, USA, 25–28 June 2012; pp. 1–12. [CrossRef]
33. Karim, M. Hasanuzzaman A Study on Mouse Movement Features to Identify User. *Sci. Res. J.* **2020**, *8*, 77–82. [CrossRef]
34. Django REST Framework. 2011. Available online: https://www.django-rest-framework.org/ (accessed on 14 June 2022).
35. INFORMS. A Measure of Distance between Time Series: Dynamic Time Warping. INFORMS. 2022. Available online: https://www.informs.org/Publications/OR-MS-Tomorrow/A-measure-of-distance-between-time-series-Dynamic-Time-Warping (accessed on 21 June 2022).
36. Morse, G. Programmatic Identification of Support/Resistance Trend Lines with Python. Medium. 2019. Available online: https://towardsdatascience.com/programmatic-identification-of-support-resistance-trend-lines-with-python-d797a4a90530 (accessed on 21 June 2022).
37. Simonyan, K.; Andrew, Z. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556. [CrossRef]
38. Keras-Applications/Vgg16.Py at Master Keras-Team/Keras-Applications. 2020. GitHub. Available online: https://github.com/keras-team/keras-applications (accessed on 21 June 2022).
39. Liu, F.; Wang, Y.; Wang, F.-C.; Zhang, Y.-Z.; Lin, J. Intelligent and Secure Content-Based Image Retrieval for Mobile Users. *IEEE Access* **2019**, *7*, 119209–119222. [CrossRef]
40. Hands-on Transfer Learning with Keras and the VGG16 Model. Available online: https://www.learndatasci.com/tutorials/hands-on-transfer-learning-keras/ (accessed on 21 June 2022).
41. Brownlee, J. Transfer Learning in Keras with Computer Vision Models. Machine Learning Mastery (Blog). 2019. Available online: https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/ (accessed on 21 June 2022).
42. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
43. Keras, T. Keras Documentation: Keras Applications. 21 June 2022. Available online: https://keras.io/api/applications/#vgg16 (accessed on 25 February 2023).

44. Cunningham, P. Dimension Reduction. In *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval, Matthieu Cord and Pádraig Cunningham*; Cognitive Technologies; Springer: Berlin/Heidelberg, Germany, 2008; pp. 91–112. [CrossRef]
45. Salgado, C.M.; Vieira, S.M. Machine Learning for Patient Stratification and Classification Part 2: Unsupervised Learning with Clustering. In *Leveraging Data Science for Global Health*; Leo Anthony, C., Maimuna, S.M., Patricia, O., Juan Sebastian, O., Kenneth, E.P., Melek., S., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 151–168. [CrossRef]
46. Penn, B.S. Using self-organizing maps to visualize high-dimensional data. *Comput. Geosci.* **2005**, *31*, 531–544. [CrossRef]
47. Moosavi, V. Sevamoo/SOMPY. Jupyter Notebook. 2014. Available online: https://github.com/sevamoo/SOMPY (accessed on 21 June 2022).
48. Gupta, R. Deeper Dive into Self-Organizing Maps (SOMs). Water Programming: A Collaborative Research Blog (Blog). 2020. Available online: https://waterprogramming.wordpress.com/2020/07/20/deeper-dive-into-self-organizing-maps-soms/ (accessed on 21 June 2022).
49. Marzouki, K.; Takeshi, Y. Novel Algorithm for Eliminating Folding Effect in Standard SOM. In *ESANN*; Citeseer: Princeton, NJ, USA, 2005; pp. 563–570.
50. Dittenbach, M.; Dieter, M.; Andreas, R. The Growing Hierarchical Self-Organizing Map. In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Como, Italy, 27 July 2000; IEEE: Piscataway, NJ, USA, 2000; pp. 15–19.
51. Deng, D.; Kasabov, N. On-line pattern analysis by evolving self-organizing maps. *Neurocomputing* **2003**, *51*, 87–103. [CrossRef]
52. Sklearn.Cluster.KMeans. Scikit-Learn. Available online: https://scikit-learn/stable/modules/generated/sklearn.cluster.KMeans.html (accessed on 22 June 2022).
53. Sklearn.Cluster.AgglomerativeClustering. Scikit-Learn. Available online: https://scikit-learn/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html (accessed on 21 June 2022).
54. Davies, D.L.; Bouldin, D.W. A Cluster Separation Measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *2*, 224–227. [CrossRef]
55. Georgios, D. Geodra/Articles. Jupyter Notebook. 2019. Available online: https://github.com/geodra/Articles/blob/85a4d13e060d45129af7b62174ea28619f4d9cf8/Davies-Bouldin%20Index%20vs%20Silhouette%20Analysis%20vs%20Elbow%20Method%20Selecting%20the%20optimal%20number%20of%20clusters%20for%20KMeans%20clustering.ipynb (accessed on 22 June 2022).
56. Aguinis, H.; Villamor, I.; Ramani, R.S. MTurk Research: Review and Recommendations. *J. Manag.* **2020**, *47*, 823–837. [CrossRef]