ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/341150268

BeCAPTCHA-Mouse: Synthetic mouse trajectories and improved bot detection

Article *in* Pattern Recognition · May 2020

DOI: 10.1016/j.patcog.2022.108643

CITATIONS
26

READS
2,859

4 authors, including:

Aythami Morales
Universidad Autónoma de Madrid
266 PUBLICATIONS **4,744** CITATIONS
SEE PROFILE

Julian Fierrez
Universidad Autónoma de Madrid
751 PUBLICATIONS **16,055** CITATIONS
SEE PROFILE

Ruben Vera-Rodriguez
Universidad Autónoma de Madrid
329 PUBLICATIONS **4,102** CITATIONS
SEE PROFILE

All content following this page was uploaded by Aythami Morales on 05 May 2020.

The user has requested enhancement of the downloaded file.

arXiv:2005.00890v1 [cs.CV] 2 May 2020

BeCAPTCHA-Mouse: Synthetic Mouse Trajectories and Improved Bot Detection

Alejandro Acien*, Aythami Morales, Julian Fierrez, Ruben Vera-Rodriguez

Biometrics and Data Pattern Analytics Lab, Universidad Autonoma de Madrid, Spain

Abstract

We first study the suitability of behavioral biometrics to distinguish between computers and humans, commonly named as bot detection. We then present BeCAPTCHA-Mouse, a bot detector based on neuromotor modeling of mouse dynamics that enhances traditional CAPTCHA methods. Our proposed bot detector is trained using both human and bot data generated by two new methods developed for generating realistic synthetic mouse trajectories: *i)* a knowledge-based method based on heuristic functions, and *ii)* a data-driven method based on Generative Adversarial Networks (GANs) in which a Generator synthesizes human-like trajectories from a Gaussian noise input. Experiments are conducted on a new testbed also introduced here and available in GitHub: BeCAPTCHA-Mouse Benchmark; useful for research in bot detection and other mouse-based HCI applications. Our benchmark data consists of 10,000 mouse trajectories including real data from 58 users and bot data with various levels of realism. Our experiments show that BeCAPTCHA-Mouse is able to detect bot trajectories of high realism with 93% of accuracy in average using only one mouse trajectory. When our approach is fused with state-of-the-art mouse dynamic features, the bot detection accuracy increases relatively by more than 36%, proving that mouse-based bot detection is a fast, easy, and reliable tool to complement traditional CAPTCHA systems.

*Corresponding author

Email addresses: alejandro.acien@uam.es (Alejandro Acien), aythami.morales@uam.es (Aythami Morales), julian.fierrez@uam.es (Julian Fierrez), ruben.vera@uam.es (Ruben Vera-Rodriguez)

Keywords:
CAPTCHA, bot detection, behavior, biometrics, mouse, neuromotor

1. Introduction

How to distinguish between human users and artificial intelligence during computer interactions is not a trivial task. This challenge was firstly discussed by Alan Turing in 1950. He investigated whether machines could show an intelligent behavior, and also how humans could be aware of these artificial behaviors. For this, he developed the famous Turing Test [1], commonly named as *The Imitation Game*, in which a human evaluator would judge natural language conversations between a human and a computer designed to generate human-like responses. The Turing Test was both influential and widely criticized and became an important concept in the artificial intelligence field [2]. However, at the epoch of Alan Turing research, the problem of machines acting like humans were commonly associated to science-fiction topics [3].

Nowadays, boosted by the last advances of machine learning technologies and worldwide connections, that science-fiction topic becomes a real hazard. As an example, bots are expected to be responsible for more than 40% of the web traffic with more than 43% of all login attempts to come from malicious botnets in the next years[1]. Malicious bots cause billionaire loses through web scraping, account takeover, account creation, credit card fraud, denial of service attacks, denial of inventory, and many other. Moreover, bots are used to influence and divide society (e.g. usage of bots to interfere during Brexit voting day [4], or to spread anxiety and sadness during the COVID-19 outbreak[2,3] through Twitter). Bots are becoming more and more sophisticated, being able to mimic human

[1] https://resources.distilnetworks.com/white-paper-reports/bad-bot-report-2019

[2] https://www.washingtonpost.com/science/2020/03/17/analysis-millions-coronavirus-tweets-shows-whole-world-is-sad/

[3] https://www.sciencealert.com/bots-are-causing-anxiety-by-spreading-coronavirus-misinformation

online behaviors. On the other hand, algorithms to distinguish between humans and bots are also getting very complex. We can distinguish two types of bot detection methods in response to those sophisticated bots:

- Active Detection. Traditionally named as CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), these algorithms determinate whether or not the user is human by performing online tasks that are difcult for software bots to solve while being easy for legitimate human users to complete. Some of the most popular CAPTCHA systems are based on: characters recognition from distorted images (text-based), class-objects identification in a set of images (image-based), and speech translation from distorted audios (audio-based).
- Passive Detection. These detectors are transparent and analyze the users behavior while they interact with the device. The last version of Google reCAPTCHA v3 replaces traditional cognitive tasks by a transparent algorithm capable of detecting bots and humans from their web behavior[4]. Other researchers [5], describe browsing behavior of web users for detection of DDoS Attacks (Distributed Denial of Service).

Although these algorithms are broadly used, they present limitations. First of all, ensuring a very accurate bot detection makes the tasks difficult to perform even for humans. Second, most of the CAPTCHA systems can be easily solved by the most modern machine learning techniques. For example, the text-based CAPTCHA was defeated by Bursztein *et al.* [6] with 98% accuracy using a ML-based system to segment and recognize the text. In [7], the authors designed an AI-based system called unCAPTCHA to break Googles most challenging audio reCAPTCHAs. Third, these algorithms process sensitive information and there are important concerns about how they comply with new regulations such as the European GDPR[5]. Fourth, the CAPTCHA systems become a great barrier to

[4] https://www.google.com/recaptcha/intro/v3.html
[5] https://complianz.io/google-recaptcha-and-the-gdpr-a-possible-conflict/

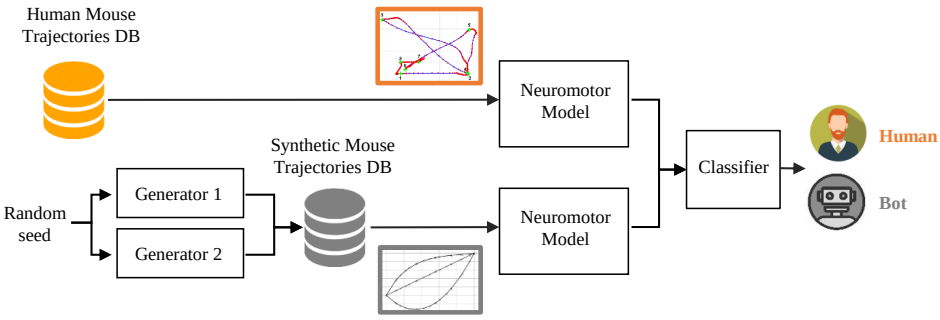

Figure 1: Architecture of BeCAPTCHA-Mouse: Neuromotor features are extracted from Human and Synthetic Mouse Trajectories. A Classifier is then trained for bot detection. The proposed Generators can be also helpful for other HCI applications.

people with visual or other impairments. Finally, the Turing Test was designed as a task in which machines had to prove they were human, meanwhile in current CAPTCHA systems humans have to prove they are not machines (e.g. *Im not a robot* from Googles). This means that the responsibility to prove the users humanity falls over human users instead of bots. At this point, there is still a large room for improvement towards reliable bot detection able to stop malicious software not bothering human users during natural web browsing.

On the other hand, biometric recognition refers to the automated recognition of individuals based on their physiological (e.g. fingerprint, face,) and behavioral (e.g. keystroke, gait) characteristics [8]. Traditionally focused on person recognition, the individual patterns obtained from biometric signals characterize the human being. Behavioral biometrics refers to those traits revealing distinctive user behaviors and mannerisms when they interact with devices [9]. Behavioral biometrics characteristics can be easily acquired with almost total transparency, being less invasive than other methodologies. The latest advances in machine learning have exposed the vulnerabilities of bot detectors [6, 7], however at the same time, these advances can be used to develop better bot detectors.

Our contributions with this work go a step forward in the bot detection field incorporating behavioral modeling and improved learning methods based on realistic synthetic samples (see Fig. 1):

Figure 2: An application example of our proposed mouse bot detection algorithm in combination with a traditional image-based CAPTCHA. While the user completes the image CAPTCHA task (cognitive challenge), our algorithm analyzes the mouse trajectories performed during the task (neuromotor challenge).

- We propose two new methods for generating realistic mouse trajectories: *i)* a knowledge-based method based on heuristic functions, and *ii)* a data-driven method based on Generative Adversarial Networks (GANs) in which a Generator synthesizes human-like trajectories from a Gaussian noise input. We demonstrate the usefulness of these synthetic trajectories to train more accurate bot detectors. These Generators can be helpful in many HCI research areas and applications.
- We propose BeCAPTCHA-Mouse, a new bot detector based on neuromotor modeling of mouse trajectories and supervised classification trained with human and synthetic data. As showed in Fig. 2, our proposed mouse detection algorithm can be added in a transparent setup and enhance traditional CAPTCHAs based on cognitive challenges, for example when you select the images in a visual CAPTCHA, or when you navigate through a website.

- We present BeCAPTCHA-Mouse Benchmark[6], the first public benchmark for mouse-based bot detection including 10,000 human and synthetic trajectories generated according to 10 different types of synthesized behaviors. The inclusion of various types of synthetic samples (both for training and testing BeCAPTCHA-Mouse) allows to train strong bot detectors. Also, it allows comprehensive evaluations under various conditions including the worst-case scenario in which bot attacks mimic human behavior using latest machine learning advances. This benchmark can be helpful for other HCI applications involving mouse dynamics beyond bot detection.

The rest of the paper is organized as follows. In Section 2 we first discuss the usage of mouse dynamics in the context of behavioral biometrics. Section 3 describes the proposed methods for generating synthetic mouse trajectories. Section 4 presents our bot detector BeCAPTCHA-Mouse. Section 5 describes our experimental framework (BeCAPTCHA-Mouse Benchmark) and presents the results obtained. Section 6 compares our BeCAPTCHA-Mouse with related state-of-the-art CAPTCHA methods. Finally, Section 7 summarizes the conclusions and future works.

2. Mouse Dynamics in the Context of Behavioral Biometrics

Human-Machine interaction generates a heterogeneous flow of data from multiple channels. This interaction generates patterns affected by: humans (e.g. attitude, emotional state, neuromotor, and cognitive abilities), sensor characteristics (e.g. ergonomics, precision), and task characteristics (e.g. easy of use, design, usefulness). Modeling the user behavior using these heterogeneous data streams is an ongoing challenge with applications in a variety of fields such as security, e-health, gaming, or education [10, 11, 12]. Among this variety of data sources, in the present paper we concentrate in behavioral biometric signals [13].

[6] https://github.com/BiDAlab/BeCAPTCHA-Mouse

	Uniq.	Univ.	Meas.	Perf.	Circ.	Acce.	Cog.	Neu.
Keystroke	**	**	***	***	**	**	**	***
Stylometry	*	*	*	*	*	*	***	*
Web-log	**	*	*** **	**	*	*	***	*
Mouse	*	**	***	***	*	***	**	***

Table 1: Biometric characteristics typically obtained in human-computer interaction. We rate each factor with * (low), ** (medium), and *** (high). Uniq = Uniqueness, Univ = Universality, Meas = Measurability, Perf = Performance, Circ = Circumvention, Acce = Acceptability, Cog = Cognitive, Neu = Neuromotor.

The literature of behavioral biometrics in the context of Human-Computer Interaction is large and includes several characteristics, e.g.: keystroking [14, 15], handwriting, touchscreen signals [16], stylometry [17, 18], and mouse dynamics [19, 20]. Each characteristic has its pros and cons, therefore, a single biometric characteristic is usually not suitable for all applications. The biometric research community has identified several factors that determine the suitability of a biometric characteristic to be used in a certain application [8].

Table 1 rates these factors for biometrics characteristics typically obtained from Human-Computer Interaction highlighting Mouse Dynamics, the focus in the present paper. Note that we added two factors related to the nature of the patterns obtained from these characteristics (Cognitive and Neuromotor patterns) with respect to the characteristics defined by [8].

Now focusing in mouse dynamics for biometrics, in [19, 20] researchers explored characteristics obtained from mouse tasks for user recognition. They analyzed up to 68 global features (e.g. duration, curvature, mean velocity) from mouse dynamics extracted during login sessions. Their results achieve up to 95% authentication accuracy for passwords with 15 digits. Besides, mouse dynamics can be combined with keystroke biometrics for continuous authentication schemes [21]. The fusion of both biometric modalities has been shown to outperform significantly each individual modality achieving up to 98% authentication accuracy [22, 23]. In [24], the authors applied the Sigma-Lognormal

Model based on the Kinematic Theory [25] to compress mouse trajectories. They suggested that mouse movements are the result of complex human motor control behaviors that can be decomposed in a sum of primal movements. In addition, in [26], the authors studied the relationship between eye gaze position and mouse cursor position on a computer screen during web browsing and suggested that there are regular patterns of eye/mouse movements associated to the motor cortex system.

3. Mouse Trajectory Synthesis: Proposed Methods

In the present paper, a mouse movement is defined by the spatial trajectory across time between two consecutive clicks, i.e., a sequence of points $\{\mathbf{x}, \mathbf{y}\}$, where $\mathbf{x} = [x_1, \ldots, x_M]$, $\mathbf{y} = [y_1, \ldots, y_M]$, and M is the number of time samples. We propose two methods for synthetically generating such mouse. A mouse trajectory is defined by two main characteristics: the shape and the velocity profile. In order to generate realistic synthetic samples, both characteristics must be considered in the generation method.

3.1. Method 1: Knowledge-based Trajectories

We generate mouse trajectories according to three different trajectory shapes (linear, quadratic, and exponential) and three different velocity profiles (constant, logarithmic, and Gaussian). We can synthesize many different mouse trajectories that mimic human movements by varying the parameters of each function. To generate a synthetic trajectory $\{\hat{\mathbf{x}}, \hat{\mathbf{y}}\}$ with M points, first we define the initial point $[\hat{x}_1, \hat{y}_1]$ and ending point $[\hat{x}_M, \hat{y}_M]$. Second, we select one of three velocity profiles: *i)* constant velocity, where the distance between adjacent points is constant; *ii)* logarithmic velocity, where the distances are gradually increasing (acceleration); and *iii)* Gaussian velocity, in which the distances first increase and then decrease when they get close to the end of the trajectory (acceleration and deceleration). Third, we generate a sequence $\hat{\mathbf{x}}$ between $\hat{x}_1$ and $\hat{x}_M$ spaced according to the selected velocity profile. The $\hat{\mathbf{y}}$ sequence is then

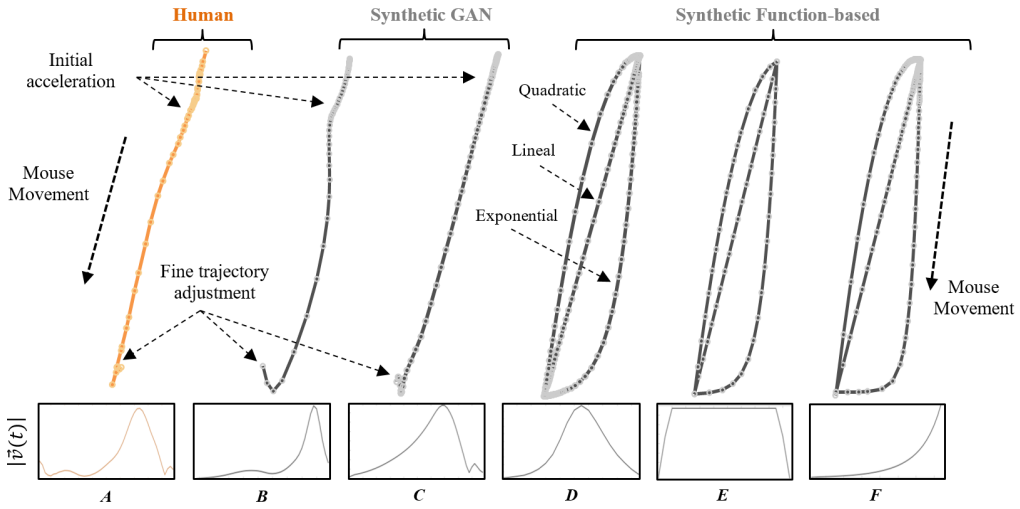

Figure 3: Examples of mouse trajectories and their velocity profiles employed in this work: A is a real one extracted from a task of the database; B and C are synthetic trajectories generated with the GAN network; D, E and F are generated with the knowledge-based approach. Note that for each velocity profile (D = Gaussian, E = constant, F = logarithmic), we include the three knowledge-based trajectories (linear, quadratic, and exponential).

generated according to the shape function. For example, for a shape defined by the quadratic function $\hat{y} = a\hat{x}^2 + b\hat{x} + c$, we fit b and c for a fixed value of a by using the initial and ending points. We repeat the process fixing either b or c. The range of the parameters $\{a, b, c\}$ explored is determined by analyzing real mouse movements fitted to quadratic functions. Linear and exponential shapes are generated similarly.

Fig. 3 (trajectories D, E, and F) shows some examples of these mouse trajectories synthesized. That figure also shows the 3 different velocity profiles considered: the 3 trajectories in E have constant velocity, F shows acceleration (the distance between adjacent samples increases gradually), and D has initial acceleration and final deceleration. We can generate infinite mouse trajectories with this approach by varying the parameters of each function.

An important factor when synthetizing mouse trajectories is the number of points (M) of the trajectory. This usually varies depending not only on the

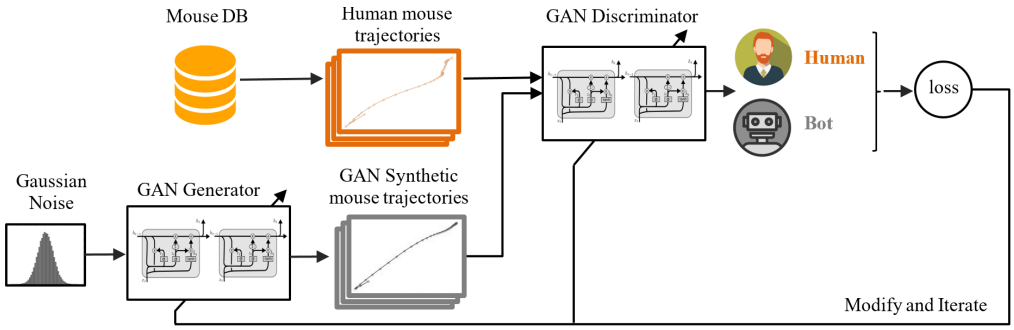

Figure 4: The proposed architecture to train a GAN Generator of synthetic mouse trajectories. The Generator learns the human features of the mouse trajectories and generate human-like ones from Gaussian Noise.

length of the trajectory, but also on the direction, because different muscles are involved when we perform mouse trajectories in different directions. To emulate this phenomenon, we calculate the mean and standard deviation of the number of points for each of the 8 mouse trajectories from the human data used in the experiments. Then, we synthetize trajectories with different number of points following a Gaussian distribution with the calculated mean and standard deviation.

3.2. Method 2: GAN-based Trajectories

For this approach we employ a GAN (Generative Adversarial Network) [27], in which two neuronal networks, commonly named Generator and Discriminator, are trained one against the other (thus the adversarial). The architecture of the GAN is depicted in Fig. 4. The aim of the Generator is to fool the Discriminator by generating fake samples (mouse trajectories in this work) very similar to the real ones while the Discriminator has to predict whether the sample comes from the real set or is a fake created by the Generator. Once the Generator is trained this way, then we can use it to synthesize mouse trajectories very similar to the human ones.

The topology employed in both Generator and Discriminator consist of two LSTM (Long Short-Term Memory) layers followed by a dense layer, very similar

to a recurrent auto-encoder. The dense layer of the Discriminator is used as a classification layer to distinguish between fake and real mouse trajectories, while the Generator employs the dense layer to build synthetic mouse trajectories. Fig. 3 shows two examples (trajectories B and C) of synthetic mouse trajectories generated with the GAN network and the comparison with a real one. We can observe high similarity between the two synthetic examples and the real one. Human mouse patterns such us the initial acceleration and the final trajectory fine correction that we discussed before are automatically learned by the GAN network and reproduced in the synthetic trajectories generated.

4. BeCAPTCHA-Mouse: Bot Detection based on Mouse Dynamics

The mouse is a very common device and its usage is ubiquitous in human-computer interfaces. Bot detection based on mouse dynamics can be therefore applied either in active or passive detectors.

In our BeCAPTCHA-Mouse bot detector we use mouse dynamics to extract neuromotor features capable to distinguish human behavior from bots (see Fig. 1). Mouse dynamics are rich in patterns capable of describing neuromotor capacities of the users. Note that we do not claim to replace other approaches (e.g. Google's reCAPTCHA) by mouse-based bot detection, our purpose is to enhance them by exploiting the ancillary information provided by mouse dynamics (see Fig. **??**).

Our proposed method for bot detection consists in characterizing each mouse trajectory with a fixed-size feature vector followed by a standard classifier. Each trajectory characterized in this way can be classified individually using standard classifiers into human or bot based on supervised training using a development groundtruth dataset. When multiple trajectories are available, standard information fusion techniques can be applied [16]. The more realistic the synthetic data used as groundtruth for training the classifier the stronger the classifier.

In our experimental work we will use Support Vector Machine classifiers, but any other standard classifier can be applied as well. The contribution and

Parameter	Description
D_i	Input pulse: covered distance
t_{0i}	Initialization time: displacement in the time axis
μ_i	Log-temporal delay
σ_i	Impulse response time of the neuromotor system
θ_{si}	Starting angle of the stroke
θ_{ei}	Ending angle of the stroke

Table 2: Sigma-Lognormal features description.

success of our BeCAPTCHA-Mouse bot detector is not in the particular classifier used, but in two other fronts (see Fig. 1): the high realism of the groundtruth data used for training our classifiers (with the methods presented in Section 3), and our proposed trajectory modeling using neuromotor features.

4.1. Neuromotor Analysis of Mouse Trajectories

By looking at typical mouse movements (see Fig. 5.a), we can observe some aspects typically performed by humans during mouse trajectories execution: an initial acceleration and final deceleration performed by the antagonist (activate the movement) and agonist muscles (opposing joint torque) [28], and a fine-correction in the direction at the end of the trajectory when the mouse cursor gets close to the click button (characterized by a low velocity that serves to improve the precision of the movement). These aspects motivated us to use neuromotor analysis to find distinctive features in human mouse movements. Neuromotor-fine skills, that are unique of human beings are difficult to emulate for bots and could provide distinctive features in order to tell humans and bots apart.

For this, we propose to model the trajectories according to the Sigma-Lognormal model [29] from the kinematic theory of rapid human movements [25]. The model states that the velocity profile of the human hand movements (mouse movements in this work) can be decomposed into primitive strokes with a Lognormal shape that describes well the nature of the hand movements ruled by the motor cortex. The velocity profile of these strokes is modeled as:

$$|\vec{v_i}(t)| = \frac{D_i}{\sqrt{2\pi}\sigma_i (t - t_{0i})} \exp\left(\frac{(\ln(t - t_{0i}) - \mu_i)^2}{-2\sigma_i^2}\right) \quad (1)$$

where the parameters are described in Table 2. The velocity profile of the entire hand movement is calculated as the sum of all these individual strokes:

$$\vec{v_r}(t) = \sum_{i=1}^{N} \vec{v_i}(t) \quad (2)$$

where N is the number of velocity strokes considered in the model. A complex action like handwriting signature or mouse movements, is a summation of these lognormals, each one characterized by the six parameters in Table 2. An example of this is shown in Fig. 5.b, where the blue line is the velocity profile $|\vec{v}(t)|$ of the above human mouse task (Fig. 5.a), which is used as the input of the Sigma-Lognormal model. The green dashed lines correspond to the individual lognormal signals $|\vec{v_i}(t)|$ generated as in [30], which describes a method to automatically estimate both N and the parameters in Table 2 from an input trajectory $|\vec{v}(t)|$. Finally, the red dotted line $|\vec{v_r}(t)|$ is the reconstruction of the original velocity profile by summing all these generated individual lognormal signals. We can observe that the reconstructed signal matches almost perfectly with the original velocity profile of the human mouse movement, suggesting the potential of the Sigma-Lognormal model to describe neuromotor mouse movements. Lognormals with a high amplitude are typically observed during the first part of the movement (agonist and antagonist activations), while smaller lognormals occur during the fine correction. The differences in lognormal sizes provide us information about the length of the trajectory (long trajectories have usually larger velocities).

The neuromotor feature set proposed for bot detection is computed from the six lognormal parameters described in Table 2. Each mouse trajectory generates N lognormal signals and each lognormal generates those 6 parameters from Table 2. For each parameter, we calculate 6 features: maximum, minimum, and mean for both halves of the trajectory. This is done because in natural mouse movements the lognormal parameters are usually very different between

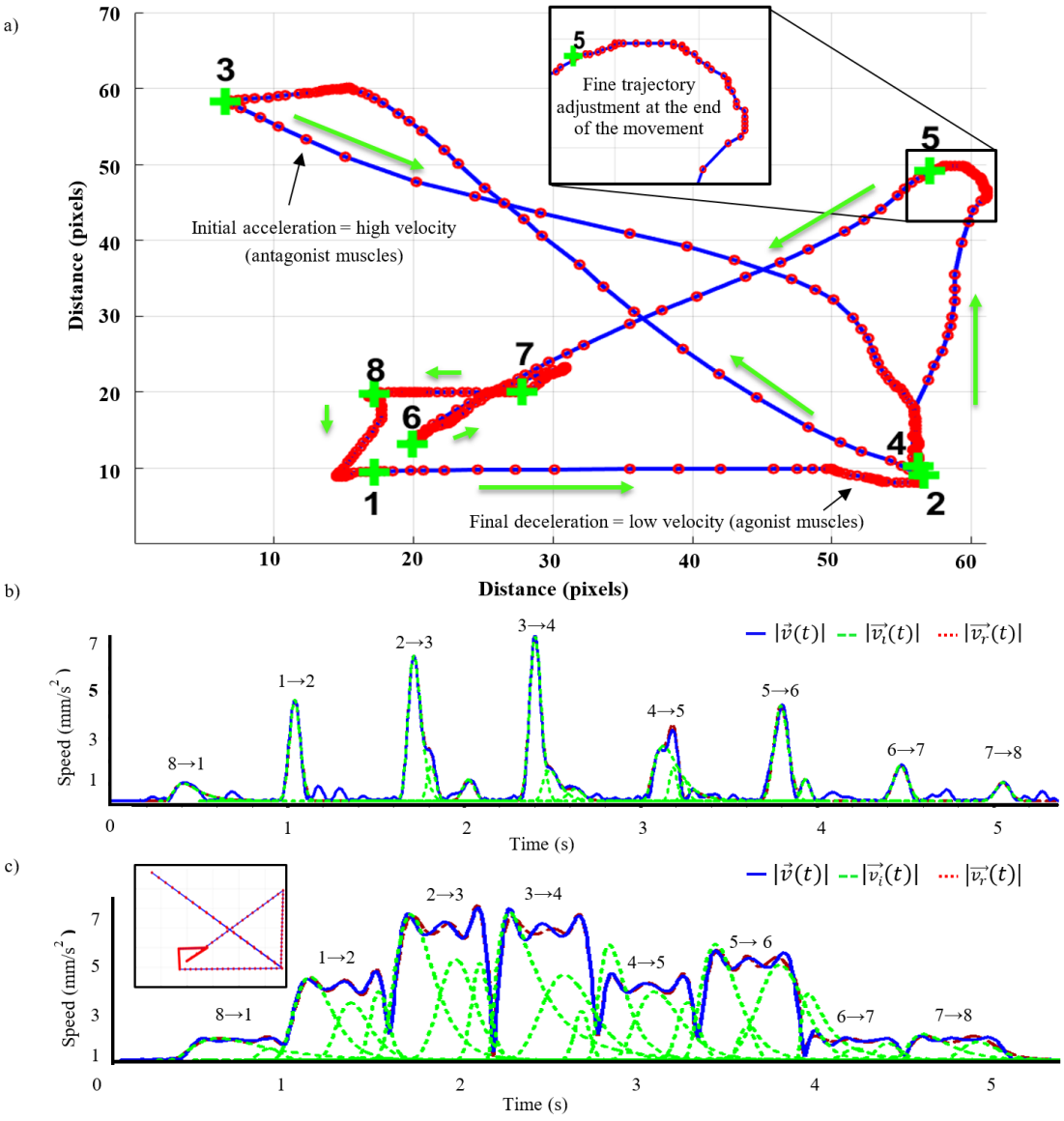

Figure 5: a) Example of the mouse task determined by 8 keypoints: the crosses represent the keypoint where the user must click, red circles are the (x,y) coordinates obtained from the mouse device, and the blue line is the mouse trajectory. b) and c) are examples of the Lognormal decomposition of a human mouse movement and a synthetic linear trajectory respectively.

both halves of a given trajectory (e.g. Fig. 5.b). Additionally, we added the number of lognormals N that each mouse trajectory generates as an additional feature. This additional feature measures the complexity of the trajectory [31], having many lognormals means that the mouse trajectory has many changes in the velocity profile while few of them usually indicates more basic trajectories. As a result, the neuromotor feature set has size 37.

5. Experiments

5.1. BeCAPTCHA-Mouse Benchmark: Database

The human mouse trajectories employed in this work were extracted from Chao *et al.* [32] database, which is comprised of more than 200K mouse trajectories acquired from 58 users who completed 300 repetitions of the task. Acquisition of data fro meach subject took between 30 days and 90 days. In each repetition, the task was to click 8 buttons that appeared in the screen sequentially. This task was repeated twice in each session. Fig. 5.a shows an example of the whole mouse movement task. Note that the buttons are placed in a particular order to generate mouse trajectories with different directions (rightwards, upwards, downwards, and oblique) and different lengths.

In the present work, we define a mouse trajectory as the mouse displacement that occurs between two click buttons. Therefore, the mouse movement task of Fig. 5.a is composed of 8 mouse trajectories. The raw data recorded during the acquisition process was: the mouse position over the screen (x,y axis position in pixels), the event (movement or click), and timestamp of the event. The experiments presented in this work are performed using a subset of the database including 35 samples (randomly chosen) from each of the 58 users available (more than 2K trajectories in total).

Fig. 5.c shows the decomposition of a synthetic knowledge-based trajectory with linear shape. We can observe the huge differences between both lognormal decompositions (the human trajectory and the synthetic one) by looking at the shape of the lognormal signals. The synthetic trajectory has wider lognormals

and they are more symmetric than the human ones. Note that the Sigma-Lognormal algorithm introduces a low-pass filter to the input signal, that is the reason why the velocity profile of the synthetic trajectory (Fig. 5.c) is a bit smoothed, but the difference between both synthetic and human velocity profiles is still patent.

5.2. Experimental Protocol

We have extracted the proposed neuromotor features from human and synthetic mouse trajectories (10K trajectories between both groups). We use an SVM (Support Vector Machine) classifier with a RBF (Radial Basis Function) kernel because of its good general performance in binary classification tasks. The experiments are divided according to the 8 real mouse trajectories present in the whole task. This means that we classify at trajectory level (i.e. the mouse trajectory performed between two consecutive click buttons) instead of classifying the whole task. This is because the task was designed to take into account different directions and length trajectories, and therefore, different muscles configurations are involved in each trajectory. In this way, we can analyze which mouse trajectories are better to discriminate between humans and bots. We train 10 different SVMs (one for each type of attack, see columns in Table 3) using both human and synthetic trajectories. For each SVM, we train the classifier by using 70% of both positive and negative samples and test with the remaining 30% (randomly chosen), each experiment was repeated 5 times and error rates were computed as the average of the 5 iterations.

The GAN network was trained using 60% of the human mouse trajectories in the database. Training details: learning rate $\alpha = 2 \times 10^{-4}$, Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, 50 epochs with a batch size of 128 samples for both Generator and Discriminator. The loss function was *binary crossentropy* for the Discriminator and *mean square error* for the Generator. The model was trained and tested using *Keras-Tensorflow*.

Trajectories		Bot: Knowledge-Based									Bot: GAN
		Linear			Quadratic			Logarithmic			
		VP = 1	VP = 2	VP = 3	VP = 1	VP = 2	VP = 3	VP = 1	VP = 2	VP = 3	
Individual trajectories	$8 \rightarrow 1$	6.7	8.3	3.3	18.7	10.0	6.0	17.3	12.4	8.04	1.8
	$1 \rightarrow 2$	1.1	5.6	2.5	6.1	5.6	8.3	8.3	3.4	10.0	3.7
	$2 \rightarrow 3$	1.1	0.8	3.9	7.2	1.7	15.7	9.4	3.9	11.1	1.3
	$3 \rightarrow 4$	1.7	2.2	6.7	5.0	2.2	13.9	5.0	2.3	12.8	0.3
	$4 \rightarrow 5$	2.2	3.9	2.5	7.8	2.2	12.8	7.2	3.4	13.3	2.5
	$5 \rightarrow 6$	1.7	4.4	6.1	3.9	1.1	15.0	3.9	5.7	11.1	1.5
	$6 \rightarrow 7$	5.0	4.4	3.3	12.2	8.9	8.9	15.0	10.3	10.6	1.5
	$7 \rightarrow 8$	4.9	7.2	7.2	10.6	11.1	9.1	13.3	16.1	17.7	0.8
All	Ours [Neuromotor]	2.3	2.9	4.1	6.1	6.5	7.7	6.4	7.6	7.9	3.9
	Baseline [33]	**0.1**	**0.2**	**0.2**	5.5	5.8	3.8	2.7	3.4	3.1	2.5
	Ours [Neuromotor]+[33]	0.2	0.4	0.3	**1.5**	**1.2**	**1.2**	**1.1**	**0.8**	**1.0**	**2.2**

Table 3: Equal Error Rate (%) in the binary classification between each of the 8 human trajectories and the synthetic ones. VP (Velocity Profile): VP = 1 constant velocity, VP = 2 initial acceleration, VP = 3 initial acceleration and final deceleration.

5.3. BeCAPTCHA-Mouse Benchmark: Results

Table 3 shows the final results for all classification schemes. The first 8 rows present the 8 trajectories derived from the movements between the 8 keypoints (plotted in Fig. 5.a). The table shows the classification errors in % (human vs bot) for the different synthetic trajectories (in columns) generated in this work. The results are presented in terms of EER (Equal Error Rate) defined as the point where the False Positive Rate and the False Negative Rate are equal.

First, comparing among the different trajectories, we can observe that the shorter ones ($8 \rightarrow 1$, $6 \rightarrow 7$, and $7 \rightarrow 8$) show higher classification errors compared to the larger ones. Short trajectories generate less neuromotor information: initial acceleration, final deceleration, and trajectory corrections are less pronounced in short trajectories. Second, logarithmic trajectory shapes achieve the worst classification performance, as we expected, because the shape of logarithmic functions fit better the human trajectories shapes. Third, the most significant parameter when synthetizing trajectories is the velocity profile.

When VP = 3 (i.e., initial acceleration and final deceleration), the synthetic trajectories are able to fool the classifier up to 17% of the times. This confirms that the velocity profile of human mouse trajectories plays and important role when describing human features in mouse dynamics. Four, the GAN Generator (last column in Table 3) results in lower classification errors compared with the knowledge-based method. This is surprising after visualizing the high similarity between human and GAN-generated trajectories (see Fig. 3 A vs B and A vs C). We interpret this result with care: on the one hand it demonstrates that out bot detection approach is powerful against realistic and sophisticate fakes, but on the other hand the GAN Generator can be improved to better fool our detector.

The last three rows in Table 3 present the results when features from all 8 trajectories are combined (each SVM is trained using features from all 8 trajectories). Additionally, we compare the performance achieved with existing approaches [33]. The feature set proposed in [33] consists of 6 global features: duration, distance, displacement, average angle, average velocity, and move efficiency (distance over displacement). The results suggest that the feature set proposed in [33] outperforms the neuromotor features proposed here only for Linear synthetic trajectories. The best performance is obtained overall with an extended set composed by both sets of features. The extended set has the best results with an average around 1% EER independently of the type of synthetic trajectory.

Finally, Table 4 shows the EER when all types of attacks are used to train and test the system. In this case, one SVM is trained using trajectories from all 8 directions and synthetic samples from all 10 types of attacks. The results show that the neuromotor feature set allows to reduce the error by 36% in comparison with the previous existing method [33]. These results demonstrate the potential of mouse dynamics features to distinguish between human and synthetic mouse movements. Additionally, we show the performance of a one class SVM classifier trained using only real samples. As can be seen, the classifier trained only with real samples was not capable to detect most of the attacks with error rates

Features	Training	
	Only Real	Real+Fake
Baseline [33]	34.7%	4.4%
Ours [Only Neuromotor]	35.6%	10.2%
Ours [Neuromotor + Baseline]	40.1%	2.8%
Error Reduction	↑ 15%	↓ 36%

Table 4: Equal Error Rate (%) in bot detection of the different feature sets for models trained with and without synthetic samples (fakes) and evaluated using human samples and fake samples. The last row shows the error reduction compared to the set proposed in [33].

over 34% either for baseline set and neuromotor features. The importance of synthetic samples is twofold: i) evaluation of bot detection algorithms under challenging attacks generated according to different methods; and ii) training better detectors to model both human and synthetic behaviors. The results in Table 4 show the potential of the synthetic samples and its usefulness to train better models capable to deal with all types of attacks.

6. BeCAPTCHA and Complementarity with the State of the Art

BeCAPTCHA-Mouse is a bot detector based on the behavior modeling of human-machine interaction. The exploitation of behavioral biometrics for bot detection is an open research line with large opportunities and challenges. These challenges include the study of new ways of interactions such as keystroke or touch [34], the applications to mobile scenarios, or the circumvention to attacks. We want to highlight that behavioral CAPTCHAs are compatible with previous CAPTCHA technologies and it could be added as a new cue to improve existing bot detection schemes in a multiple classifier combination [16] (see Fig. 6).

Table 5 shows some of the main features of different existing CAPTCHA methods. As we commented in the introduction section, most of them have been defeated by machine learning algorithms. In fact, the last version of the Google CAPTCHA, named reCAPTCHAv3, that measures mouse dynamics

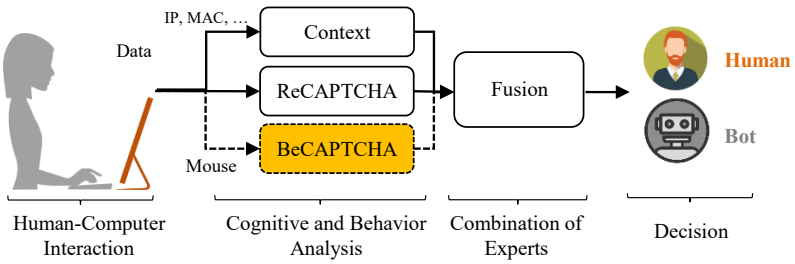

Figure 6: Block diagram of multimodal bot detection. The response of the bot detector is a combination of responses from different experts. The bot detector proposed in this work can be used independently or in combination with existing bot detectors.

and web browsing interactions between the user and the web site to decide whether the user is a bot or not, was recently hacked in [35] by synthetizing mouse trajectories using reinforcement learning techniques. The main problem of these CAPTCHA methods is that they only measure cognitive human skills (e.g. character recognition from distorted images, class-objects identification in a set of images, or speech translation from distorted audios). Trying to ensure a very accurate bot detection makes these CAPTCHAs difficult to perform even for humans. The main goal of our proposed method is to focus more on human behavioral skills rather than on cognitive ones. Neuromotor skills reveal human features useful for bot detection just with simple mouse trajectories. To the best of our knowledge, there are only a very limited number of works using mouse biometrics for bot detection. The most related to our research are [33] and [35]. In [35] they synthetize mouse trajectories over a grid to hack the Google reCAPTCHA v3 algorithm, and in [33] they extract global features (e.g. duration, average speed, displacement) from mouse and keystroke patterns to conduct a case study in the detection of blog bots for online blogging systems. While previous work in mouse dynamics ([19, 20, 33]) focused on basic cues like duration or average speed, in this work we go a step forward by focusing on the analysis of entire mouse trajectories, using the Sigma-Lognormal model to extract human features that characterizes better human behaviors.

Method	Cog.	Beh.	**Usability**	**Security**
Audio CAPTCHA	***	*	*	*
Image CAPTCHA	***	*	*	*
Text CAPTCHA	***	*	*	*
reCAPTCHA v3	*	**	***	**
Our method	**	**	***	***

Table 5: Characteristics of several CAPTCHA methods. We rate each factor as low (*), medium (**) and high (***). Cog = Cognitive, Beh = Behavioral.

7. Conclusions and Future work

We have explored behavioral biometrics for bot detection during human-computer interaction. In particular, we have analyzed the capacity of mouse dynamics to describe human neuromotor features. Our conclusions in comparison to state-of-the-art works suggest that there is unexploited potential of mouse dynamics as a behavioral biometric for tasks such as bot detection.

In concrete, we have proposed BeCAPTCHA-Mouse, a bot detection algorithm based on mouse dynamics, and a related benchmark[7], the first one public for research in bot detection and other mouse-based research areas including HCI, security, and human behavior.

Additionally, we have proposed and studied two new methods for generating synthetic mouse trajectories of varying level of realism. These generators are very useful both training stronger bot detectors, and evaluating them in comprehensive and worst case scenarios. These generators are also valuable for related research problems beyond bot detection involving mouse dynamics.

In our experiments we have observed the main features of human mouse trajectories (e.g. initial acceleration, final deceleration, and fine trajectory correction). Based on that we have developed a neuromotor feature representation using the Sigma-Lognormal model [25, 29]. Using the proposed neuromotor fea-

[7] https://github.com/BiDAlab/BeCAPTCHA-Mouse

ture representation and training standard classifiers making use of the proposed synthetic mouse trajectories, we have been able to discriminate between humans and bots with up to 93% of accuracy, even with bots of high realism, and only one mouse trajectory as input (between two consecutive clicks). This proves the potential of mouse dynamics for Turing tests.

As future work, we aim at improving the neuromotor feature set by calculating secondary features inferred from the main ones. Also, we propose to combine both synthesis methods by using the knowledge-based trajectories as the input of the GAN model instead of Gaussian noise. This technique could generate more sophisticate and human-like trajectories. Finally, in this paper we only considered mouse trajectories acquired from mouse devices. We also propose to analyze mouse-pad trajectories normally performed when using laptops as another line of research.

Acknowledgements

This work has been supported by projects: IDEA-FAST (H2020-IMI2-2018-853981), PRIMA (H2020-ITN-2019-860315), TRESPASS-ETN (H2020-ITN-2019-860813), BIBECA (RTI2018-101248-B-I00 MINECO/FEDER), and BioGuard (Ayudas Fundacin BBVA a Equipos de Investigacin Cientfica 2017). Spanish Patent Application P202030066.

References

[1] A. M. Turing, Computing Machinery and Intelligence, Springer Netherlands, Dordrecht, 2009, pp. 23–65.

[2] A. P. Saygin, I. Cicekli, V. Akman, Turing test: 50 years later, Minds and Machines 10 (2000) 463–518.

[3] J. Svilpis, The science-fiction prehistory of the turing test, Science Fiction Studies 35 (3) (2008) 430–449.

[4] Y. Gorodnichenko, T. Pham, O. Talavera, Social media, sentiment and public opinions: Evidence from #Brexit and #USElection, Working Paper 24631, National Bureau of Economic Research (2018).

[5] Y. Xie, S.-Z. Yu, A large-scale hidden semi-markov model for anomaly detection on user browsing behaviors, IEEE/ACM Trans. Netw. 17 (2009) 54–65.

[6] E. Bursztein, M. Martin, J. Mitchell, Text-based CAPTCHA strengths and weaknesses, in: Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 11, Association for Computing Machinery, New York, NY, USA, 2011, pp. 125–138.

[7] K. Bock, D. Patel, G. Hughey, D. Levin, unCaptcha: A low-resource defeat of reCaptcha's audio challenge, in: 11th USENIX Workshop on Offensive Technologies (WOOT 17), USENIX Association, Vancouver, BC, 2017.

[8] A. K. Jain, K. Nandakumar, A. Ross, 50 years of biometric research: Accomplishments, challenges, and opportunities, Pattern Recognition Letters 79 (2016) 80–105.

[9] J. Fierrez, A. Pozo, M. Martinez-Diaz, J. Galbally, A. Morales, Benchmarking touchscreen biometrics for mobile authentication, IEEE Trans. on Information Forensics and Security 13 (11) (2018) 2720–2733.

[10] H. J. Escalante, H. Kaya, A. A. Salah, S. Escalera, Y. G;ltrk, U. Gl, X. Bar, I. Guyon, J. C. S. Jacques, M. Madadi, S. Ayache, E. Viegas, F. Gurpinar, A. S. Wicaksana, C. Liem, M. A. J. Van Gerven, R. Van Lier, Modeling, recognizing, and explaining apparent personality from videos, IEEE Transactions on Affective Computing.

[11] E. Nosakhare, R. Picard, Toward assessing and recommending combinations of behaviors for improving health and well-being, ACM Trans. Comput. Healthcare 1 (1).

[12] H. Shrobe, D. L. Shrier, A. Pentland, Behavioral biometrics, in: New Solutions for Cybersecurity, 2018, pp. 365–377.

[13] A. Alzubaidi, J. Kalita, Authentication of smartphone users using behavioral biometrics, IEEE Communications Surveys Tutorials 18 (3) (2016) 1998–2026.

[14] C. C. Tappert, S.-H. Cha, M. Villani, R. S. Zack, A keystroke biometric system for long-text input, International Journal of Information Security and Privacy 4 (1) (2010) 32–60.

[15] A. Acien, J. V. Monaco, A. Morales, R. Vera-Rodriguez, J. Fierrez, Typenet: Scaling up keystroke biometrics, arXiv:2004.03627.

[16] J. Fierrez, A. Morales, R. Vera-Rodriguez, D. Camacho, Multiple classifiers in biometrics. part 2: Trends and challenges, Information Fusion 44 (2018) 103–112.

[17] J. C. Stewart, J. V. Monaco, S. Cha, C. C. Tappert, An investigation of keystroke and stylometry traits for authenticating online test takers, in: Proceedings of the 2011 International Joint Conference on Biometrics (IJCB), 2011, pp. 1–7.

[18] H. Locklear, S. Govindarajan, Z. Sitov, A. Goodkind, D. G. Brizan, A. Rosenberg, V. V. Phoha, P. Gasti, K. S. Balagani, Continuous authentication with cognition-centric text production and revision features, in: Proceedings of the IEEE International Joint Conference on Biometrics, 2014, pp. 1–8.

[19] A. A. E. Ahmed, I. Traore, A new biometric technology based on mouse dynamics, IEEE Transactions on Dependable and Secure Computing 4 (3) (2007) 165–179.

[20] H. Gamboa, A. L. N. Fred, A. K. Jain, Webbiometrics: User verification via web interaction, in: Proceedings of the 2007 Biometrics Symposium, 2007, pp. 1–6.

[21] T. Sim, S. Zhang, R. Janakiraman, S. Kumar, Continuous verification using multimodal biometrics, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (4) (2007) 687–700.

[22] S. Mondal, P. Bours, A study on continuous authentication using a combination of keystroke and mouse biometrics, Neurocomputing 230 (2017) 1–22.

[23] K. O. Bailey, J. S. Okolica, G. L. Peterson, User identification and authentication using multi-modal behavioral biometrics, Computers & Security 43 (2014) 77–89.

[24] D. Martín-Albo, L. A. Leiva, J. Huang, R. Plamondon, Strokes of insight, Inf. Process. Manage. 52 (6) (2016) 989–1003.

[25] R. Plamondon, A kinematic theory of rapid human movements, Biological Cybernetics 72 (4) (1995) 295–30.

[26] M. C. Chen, J. R. Anderson, M. H. Sohn, What can a mouse cursor tell us more? correlation of eye/mouse movements on web browsing, in: Proceedings of the CHI 01 Extended Abstracts on Human Factors in Computing Systems, CHI EA 01, New York, NY, USA, 2001, pp. 281–282.

[27] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, Cambridge, MA, USA, 2014, pp. 2672–2680.

[28] M. Djioua, R. Plamondon, A new algorithm and system for the characterization of handwriting strokes with delta-lognormal parameters, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (11) (2009) 2060–2072.

[29] A. Fischer, R. Plamondon, Signature verification based on the kinematic theory of rapid human movements, IEEE Transactions on Human-Machine Systems 47 (2) (2017) 169–180.

[30] O. Loyola-González, R. Monroy, M. A. Medina-Pérez, B. Cervantes, J. E. Grimaldo-Tijerina, An approach based on contrast patterns for bot detection on web log files, in: I. Batyrshin, M. d. L. Martínez-Villaseñor, H. E. Ponce Espinosa (Eds.), Advances in Soft Computing, Springer International Publishing, Cham, 2018, pp. 276–285.

[31] R. Vera-Rodriguez, R. Tolosana, J. Hernandez-Ortega, A. Acien, A. Morales, J. Fierrez, J. Ortega-Garcia, Modeling the complexity of signature and touch-screen biometrics using the lognormality principle, in: R. Plamondon, A. Marcelli, M. A. Ferrer (Eds.), The Lognormality Principle and its Applications, World Scientific, 2019.

[32] C. Shen, Z. Cai, X. Guan, R. Maxion, Performance evaluation of anomaly-detection algorithms for mouse dynamics, Computers & Security 45 (2014) 156–171.

[33] Z. Chu, S. Gianvecchio, H. Wang, Bot or Human? A Behavior-Based Online Bot Detection System, Springer International Publishing, Cham, 2018, pp. 432–449.

[34] A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez, I. Bartolome, BeCAPTCHA: detecting human behavior in smartphone interaction using multiple inbuilt sensors, in: Proceedings of the AAAI Workshop on Artificial for Cyber Security (AICS), 2020.

[35] I. Akrout, A. Feriani, Akrout, Hacking google reCAPTCHA v3 using reinforcement learning, in: Conference on Reinforcement Learning and Decision Making, 2019.

View publication stats