Common Evaluation Pitfalls in Touch-Based Authentication Systems

Martin Georgiev
University of Oxford
Oxford, UK
martin.georgiev@cs.ox.ac.uk

Simon Eberz
University of Oxford
Oxford, UK
simon.eberz@cs.ox.ac.uk

Henry Turner
University of Oxford
Oxford, UK
henry.turner@cs.ox.ac.uk

Giulio Lovisotto
University of Oxford
Oxford, UK
giulio.lovisotto@cs.ox.ac.uk

Ivan Martinovic
University of Oxford
Oxford, UK
ivan.martinovic@cs.ox.ac.uk

ABSTRACT

In this paper, we investigate common pitfalls affecting the evaluation of authentication systems based on touch dynamics. We consider different factors that lead to misrepresented performance, are incompatible with stated system and threat models or impede reproducibility and comparability with previous work. Specifically, we investigate the effects of (i) small sample sizes (both number of users and recording sessions), (ii) using different phone models in training data, (iii) selecting non-contiguous training data, (iv) inserting attacker samples in training data and (v) swipe aggregation. We perform a systematic review of 30 touch dynamics papers showing that all of them overlook at least one of these pitfalls. To quantify each pitfall's effect, we design a set of experiments and collect a new longitudinal dataset of touch dynamics from 470 users over 31 days comprised of 1,166,092 unique swipes. We make this dataset and our code available online. Our results show significant percentage-point changes in reported mean EER for several pitfalls: including attacker data (2.55%), non-contiguous training data (3.8%), phone model mixing (3.2%-5.8%). We show that, in a common evaluation setting, cumulative effects of these evaluation choices result in a combined difference of 8.9% EER. We also largely observe these effects across the entire ROC curve. Furthermore, we validate the pitfalls on four distinct classifiers - SVM, Random Forest, Neural Network, and kNN. Based on these insights, we propose a set of best practices that, if followed, will lead to more realistic and comparable reporting of results in the field.

1 INTRODUCTION

Touch dynamics systems use distinctive touchscreen gestures for authentication. These interactions include both common gestures like swipes and scrolls and more advanced ones like pinch-and-zoom. Touch dynamics have been proposed as a way to improve the security of login-time authentication mechanisms and to enable continuous authentication while a device is being used. The field has been growing rapidly since the first papers were published in 2012, with 30 papers collecting unique swipe-and-scroll datasets published so far.

Despite the growth in the field, no standard set of methods has been established to enable comparison between published work and transition to real-world deployment. While authors largely report the Equal Error Rate (EER) as a metric of average system performance, there are vast differences in methodological choices when evaluating systems on a static dataset. The goal of this paper is to identify these methodological choices, investigate how common they are in published work, and quantify their effect on reported system performance. These steps are crucial to enable fair comparisons between papers, ensure reproducibility of results and obtain results that are compatible with a real-world system- and threat model.

Through our analysis of the existing work, we identify six pitfalls where design flaws in the experiment, data collection, or analysis impede comparability or lead to unrealistic results. To examine the impact of each of these pitfalls on a touch dynamics system we collect our own longitudinal large-scale dataset of swipes. Specifically, we investigate the effects of sample and model size, mixing different phone models in the analysis, using non-contiguous training data, including attacker data in training, using arbitrary aggregation windows, and the implications of code and data availability. We quantify the effect of each pitfall with their effect on the system equal error rate, showing that pitfalls lead to conspicuous changes in the resulting performance. The dataset and code from our study are openly accessible to advance the field further.

In this study we make the following key contributions:

- We identified six evaluation pitfalls: small sample size, phone model mixing, selecting non-contiguous training data, including attacker samples, swipe aggregation, and code/dataset availability. We conducted a systematic analysis of the touch-based authentication literature, showing that all published studies overlook at least one of the pitfalls.
- We quantified the effects stemming from these pitfalls in terms of resulting EER; to do so we collected a new 470-user touch dynamics dataset comprised of daily interactions over 31 days. The dataset and our code are available online.[1]
- We outlined a set of best practices to avoid the identified pitfalls. These practices include both recommendations for experimental design and methods and also recommendations to allow for reproducibility and comparability of results in the field.

2 COMMON EVALUATION PITFALLS

In this section, we present our identified evaluation pitfalls in touch authentication systems.

[1] https://github.com/ssloxford/evaluation-pitfalls-touch

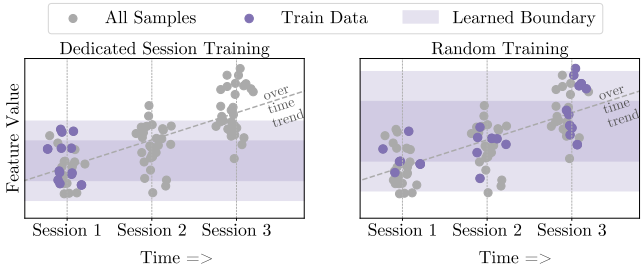


Figure 1: Examples of training data selection approaches. The "dedicated sessions" method samples data from self-contained sessions and does not posses information from future ones. The "random" method takes training samples from all session aiding in generalization although it does not represent a realistic authentication scenario.

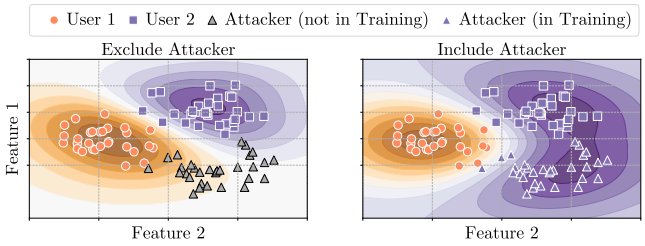


Figure 2: Visualization of the difference between attacker modeling approaches. The "include attacker" model creates a better boundary between legitimate and invalid data but it does not represent a realistic authentication scenario as specific attacker data is rarely available at the time of model creation.

P1: Small sample size. Sample size can refer both to the number of users in a study and the amount of data collection sessions recorded per user. Due to various experimental limitations, often touch authentication methods are evaluated on limited amounts of data, with a median of ~40 distinct users and two data collection sessions. Nevertheless, the accuracy of the measured performance may benefit from a larger number of users. In fact, sampling negative training data from larger pools of users can lead to differences in the performance of the recognition model, affecting the mean system performance. On the other hand, collecting longitudinal data is also necessary to estimate the effect of changing user behavior over time, as this may change across different sessions. These sample size effects are non-trivial to measure and hinder a robust generalization of results found on smaller samples.

P2: Phone model mixing. Many studies in the field perform data collection on multiple distinct device models. This can be a result of convenience (especially in remote studies) or an attempt to demonstrate system performance on different hardware. While phone models might look similar, slightly different specifications cause fundamental differences when devices are used to collect swipes. These differences are caused by various factors, including the shape of the phone, its resolution, how it is held, touchscreen sampling rate, and the value range of its pressure and area sensors. In general, an attacker would use the same phone model as their victim as they use the same physical device in an in-person attack. Mixing phone models in testing violates this requirement as attackers and victims use different device models. It is worth noting that this pitfall does not apply in the case of remote authentication where the attacker can send data from any device model.

P3: Non-contiguous training data selection. In practice, a biometric authentication system has an enrollment (training) phase which precedes the use of the system (or its evaluation). However, when using the randomized training data selection method, swipes are randomly sampled from the whole user data as shown in Figure 1 (right). This does not resemble how a deployed system works, as it essentially evaluates the system by testing on samples from the past. As a consequence, randomized training data selection leads to biased performance estimation.

P4: Attacker data in training. While there are several ways to design an authentication method, a common approach is to use a binary classifier that discriminates between legitimate and non-legitimate user samples. In this case, the negative samples (non-legitimate) are generally gathered from the available pool of users, the same user pool is then used to test the system recognition rates. However, most stated threat models rule out the possibility that the classifier was trained with negative training data belonging to an attacker: attacker samples should be *unknown*. Figure 2 illustrates this problem: including the attacker samples in the training data provides a significant benefit against attacks compared to what happens when the attacker is excluded from training. This property has been initially addressed in [11], where it is shown that it artificially reduces the zero-effort attack success rates. The inclusion of an attacker in training data is incompatible with a realistic threat model. It is important to clarify that attacker data we use to delineate the negative class consists of legitimate swipes of other users. While active attacks are interesting to examine, we limit our analysis to zero-effort attackers.

P5: Aggregation window size. Intuitively, the use of multiple swipes when evaluating a particular model leads to an increase in performance [17, 19, 20, 32, 44]. While aggregating multiple swipes for an authentication decision is a legitimate approach in general (e.g., it mitigates occasional erratic behavior and improves recognition), it has two important drawbacks. Firstly, it impedes straightforward comparison between different approaches when the aggregation window size is different. Secondly, in a realistic threat scenario, it allows the attacker a non-negligible time to perform their attack, as the anomalous attacker behavior will only be identified after a certain number of swipes (depending on the aggregation window size).

P6: Dataset and code availability. Datasets and codebases of touch-based authentication systems are rarely made publicly available. This is a major impediment to reproducibility and progress in the field. Sharing datasets would enable researchers to reliably separate the effects of different models from those of the collected data. Sharing the code used to obtain the results is especially important in light of the pitfalls investigated in this paper: oftentimes unstated assumptions are made which are not trivial to spot.

3 RELATED WORK

The focus of our work is on mobile continuous authentication systems based on swiping and scrolling behavior. While our work

concentrates on the use of swipes as the most widespread touch method, there are other types of touch gestures used for authentication (e.g., "pinch to zoom" [40], screen taps [45]). In this paper, we consider *swipes* and *scrolls* - horizontal and vertical displacements on a touch-capacitive display done using a single finger.

3.1 Background

Origin of touch-based authentication. Feng et al. developed one of the earliest systems in touch-based continuous authentication on smartphones [13]. Soon after, other systems solely based on the data provided by the phone were developed [17, 20, 25]. Many hybrid approaches for touch-based authentication have also been proposed. For instance, some research includes sensor data coming from the accelerometer and gyroscope [22, 39]. Deb et al. include 30 different modalities including GPS and magnetometer [10] and Rahul et al. have even taken into account the power usage of the device [26].

Data collection modalities. There are varying approaches for data collection in touch-based authentication. Frank et al. use text-reading to collect vertical scrolls and a "spot the difference" game to gather horizontal swipes [17]. Similarly, Antal et al. use text reading and image gallery tasks [5]. Others include social media interactions [26], zooming on pictures [15] and questionnaires [32]. Buschek et al. evaluate the influence of GUI elements and hand postures on the performance of touch dynamic systems [9]. In order to analyze the time stability of the biometric, some recent studies collect data over multiple sessions or days. Watanabe et al. specifically look into the long-term performance of touch-based systems by collecting data over 6 months [38]. They demonstrate promising results for the time-stability of the biometric. While the data from some experiments is collected in a restricted environment during lab sessions, Feng et al. [15] recruited 100 users to use their data collection application over the course of 3 weeks to provide a more realistic environment when performing everyday tasks.

Feature extraction and classification modalities. Most feature extraction methods in touch authentication systems focus on describing the geometrical attributes of swipes such as coordinates, duration, acceleration, deviation, and direction [17, 20]. Zhao et al., however, use a method to convert the stroke information into an image that can be used for statistical feature model extraction [44]. There is a vast variability in the classification approaches in touch-based authentication. Some studies have focused on systematizing and comparing knowledge within the field. Fierrez et al. [6] analyze and compare recent efforts in the field in terms of datasets, classifiers, and performance. Serwadda et al. compare the most common machine learning algorithms in the context of touch-based authentication [32]. The studies suggest that Support Vector Machine (SVM) and Random Forest perform the best for touch-based tasks. Fierrez et al. provide insights into model and design choice performance by benchmarking open-access datasets [16]. They find that landscape phone orientation and horizontal gestures prove to be more stable and discriminative.

Performance and metrics. The difference in data collection and classification approaches leads to significant variability in the results reported in the field, with authors claiming EERs between 0% [8, 17] and 22.1% [22] Studies also vary in their evaluation metrics as results are reported in False Acceptance Rate (FAR), False Rejection Rate (FRR), Equal Error Rate (EER), Receiver Operating Characteristics (ROC) curve, and Accuracy. While it has been argued that EER does not adequately describe systematic errors [12], it is generally accepted as a good measure of average system performance. Furthermore, [34] argues the importance of considering the ROC curve for performance as the EER metric could be misleading depending on TPR (True Positive Rate) and FPR (False Positive Rate) system requirements. In this paper, we abstract from the variety of experimental choices outlined in this section and investigate fundamental effects of evaluation pitfalls on the EER and ROC curve.

3.2 Prevalence of evaluation pitfalls

To check how prevalent the pitfalls are, we analyzed the touch-authentication literature. We report an overview of our findings on 30 studies from the last decade, each of the studies introduces a new touch-based dataset in Table 1. We only selected studies with experiments containing natural swiping behavior such as navigating through specific tasks. We did not consider studies that only rely on mobile keystroke dynamics, sensors, tapping, and one-time gestures for authentication. Patterns that emerge are discussed throughout the paper. Table 1 shows that all of the studies included in the table are subject to at least one of the pitfalls described in Section 2.

Our set of studies have a close to equal split in their study environment, with 15 studies done in a lab and 13 remotely – the collection environment was unclear for the 2 remaining studies. We find that the median number of participants is 40, who complete a median of 2 sessions. This relatively low number of median sessions is concerning and we analyze the impact of this (P1) in section 7. Seven of the studies hand out devices to participants for a period of time without specific instructions on how often to use them, meaning that the precise number of sessions is not known.

Of our analyzed studies, 28% mix device models in their data collection and do not discuss splitting them in the evaluation, falling into P2.

Likewise, 30% of the studies do not clearly explain the way they select their training and testing data, with a further 18% using a randomized approach to select data, and are thus snared by P3. For those that do not explain their selection process, the code is also not shared, making it impossible to know how the selection was performed.

In terms of attacker modeling, an overwhelming majority (80%) of the studies investigated use an unrealistic attacker modeling approach and include attacker data into the training set, falling victim to P4. A much smaller number of studies succumb to P5, with 17% reporting their results only on the analysis of an aggregation group of more than one swipe, hindering comparability across studies.

P6 also captures many works, with only 8 studies (27%) sharing their datasets upon publication, two of which no longer have functional web pages. Furthermore, none of the studies we examined share a complete codebase of their work. One study, [17], does share the feature extraction code files but not the rest of the analysis.

Table 1: Data collection and analysis choices in touch dynamics studies. ● denotes that the study fulfills the column recommendation (i.e., does not fall into the evaluation pitfall) and ○ denotes that it does not, ? means that the information was not shared or it is unclear from the paper, — means not applicable and ◐ in the last column means that the code or dataset is no longer available through the provided url (accessed on 14 April 2021)). The "Cont. (Period)" Sessions label indicates that the phone was given to the users for a period of time without specific instructions on how often to use it. The "Single Device Model" column marks whether the analysis separates data belonging to distinct phone models (even if the data collection included various phone models).

		P1		P2	P3	P4	P5	P6
Study (Publication Year)	Environment	Users	Sessions	Single Device Model	Contiguous Training Data	Exclude Attacker From Training	Single Gesture Analysis Available (Aggregation Sizes)	Dataset / Code Availability
[17] (2012)	Lab	41	3	○	●	○	● (1-20)	● / ○
[14] (2012)	Lab	40	1	●	?	○	● (1-9)	○ / ○
[20] (2013)	Remote	75	Cont. (?)	●	●	●	○ (2-20)	○ / ○
[8] (2013)	Remote	100	Cont. (?)	?	?	○	● (1-30)	○ / ○
[32] (2013)	Lab	190	2	●	●	○	○ (10)	◐ / ○
[18] (2014)	Remote	32	Cont. (5-10 weeks)	○	●	○	● (1)	○ / ○
[15] (2014)	Remote	23	Cont. (3 weeks)	○	○	●	● (1-10)	○ / ○
[31] (2014)	Lab	20	1	●	○	●	● (1)	○ / ○
[44] (2014)	Lab	78	6	●	?	?	● (1-7)	○ / ○
[40] (2014)	Lab	32	1	●	●	●	● (1,3,5)	◐ / ○
[43] (2015)	Lab	50	1	●	●	○	● (1-19)	○ / ○
[28] (2015)	Lab	20	1	●	?	?	● (1)	○ / ○
[5] (2015)	Remote	71	4	○	?	○	● (1-20)	● / ○
[42] (2015)	?	14	1	●	?	○	● (1-15)	○ / ○
[36] (2015)	Remote	22	30	●	●	○	—	○ / ○
[27] (2015)	Lab	73	2	●	●	○	● (1)	○ / ○
[33] (2016)	Lab	24	3	●	●	○	● (1-20)	○ / ○
[7] (2016)	Lab	40	1	●	○	○	● (1-5)	○ / ○
[23] (2016)	Remote	48	Cont. (2 months)	●	●	○	○ (2-16)	● / ○
[19] (2016)	Remote	28	7	○	●	○	○ (4)	○ / ○
[3] (2017)	?	40	1	○	?	?	● (1,5,11)	○ / ○
[38] (2017)	Remote	40	Cont. (6 months)	●	●	○	● (1)	○ / ○
[37] (2017)	Lab	20	8	●	○	●	● (1)	○ / ○
[4] (2017)	Lab	20	1	●	●	○	● (1-5)	○ / ○
[24] (2018)	Remote	48	20	●	●	○	● (1)	○ / ○
[35] (2019)	Lab	31	8	●	?	○	○ (5)	● / ○
[29] (2019)	Remote	2218	1 - 7619	○	—	—	—	● / —
[41] (2019)	Remote	45	Cont. (2 weeks)	●	?	○	● (1)	○ / ○
[30] (2020)	Lab	30	1	●	○	○	● (1)	○ / ○
[2] (2021)	Remote	600	5	○	—	—	—	● / —
Ours	Remote	470	31	Both	Both	Both	● (1-20)	● / ●

Recent studies have gathered large amounts of data by making collection apps available on public app stores [2, 29]. This is a step in the right direction in terms of dataset sizes but presents other challenges. For instance, in the case of [29] there is data from 2218 users collected on 2418 different devices and in [2] there is data from 600 users on 278 distinct devices. There is likely a large variation in the unique device models used as well, especially considering the large fragmentation of the Android ecosystem. Furthermore, multiple people may perform the tasks on the same account (e.g. a parent giving a child to play the game).

4 STUDY DESIGN

We designed our data collection experiment to enable us to thoroughly measure the effects of each of the pitfalls described in Section 2. As a consequence, we have a few notable differences from previous datasets. We collected all data remotely on a carefully constrained set of devices. Furthermore, we obtained data from 470 participants (well above the median of 40) and collected data of up to 31 sessions (compared to the median of 2). In the remainder of this section, we discuss the designs of the key parts of our data collection experiment.

4.1 Remote collection

Remote data collection provides two major benefits. Firstly, it allows for the collection of large amounts of data which is impractical for a lab study due to the difficulty of recruiting participants with particular qualities at scale. Furthermore, external factors such as the COVID-19 pandemic may prevent lab studies altogether, leaving remote collection as the only viable option. For our study, we utilized Amazon Mechanical Turk (MTurk) - a popular crowdsourcing platform, where workers perform Human Intelligence Tasks (HITs) in exchange for payment. The platform gives access to a large population of potential subjects and allows for targeting by age, gender, and other demographic criteria.

We created an MTurk HIT, which described the requirements and details of the study and guided the subjects to install the data

collection app which was distributed through TestFlight - an online service for over-the-air installation on the iOS platform, which does not allow the general public to install the application. The HIT also contained the participant information sheet, as required by our institutional review board. This study received ethical approval.

Application Onboarding. Upon opening the application, users were required to complete a consent form and provide demographic information as they would in a lab study. Users were then required to complete their first pair of tasks once. This established a connection between MTurk and the application, providing users with the first payment, and allowing payments to be automatically generated for subsequent completions of the task.

Study Duration. Within the study, participants were either invited to participate for 7 or 31 days. Each day participants were prompted with a notification (if they allowed notifications from the application) to complete the task at 9 am, and again at 7 pm if the task had not yet been completed that day. Not all users, however, completed their tasks consistently as further discussed in Appendix D.

4.2 Devices

We selected the iOS platform to carry out our data collection efforts in order to ensure the consistency of hardware and software throughout experimentation. The other major mobile operating system, Android, includes a much higher number of device models with varying screen sizes and sensors making it impractical for our analysis. Moreover, the majority of Android devices approximate their reported touch pressure values by considering the size of the touchpoint while the iPhone models we have chosen support "3D touch" - a true pressure sensor built into the screen of the devices. Due to these restrictions, we have narrowed down our efforts to the nine devices shown in Appendix E.

These design choices left us with a large number of users using a limited amount of models but still let us make a comparison in terms of phone size, resolution, and even hardware differences. To our knowledge, there is only one other paper [43] in the field which focuses on iOS devices for touch-based authentication and the dataset is not publicly available. While we have placed specific restrictions on our data collection and experimentation, the dataset can be used for developing systems beyond the specifics of this study.

4.3 Application

To facilitate our study we developed an iOS application that collects touch and sensor data as users perform common smartphone interactions. We collected coordinates and pressure data for each user interaction with the screen at the maximum rate of 60Hz. Furthermore, we also recorded the accelerometer and gyroscope data at their maximum frequency of 100Hz.

The application required users to complete two tasks: a social media style task and an image gallery task. The design and intention of these tasks are described in Section 4.4. We optimized the number of rounds of each task to equalize the completion time and the number of swipes and scrolls collected per task. Both tasks were intended to be completed with the phone in a vertical position, and thus we did not allow a change in the layout when the device was rotated. The application home page included elements such as completion streak and earning potential in order to increase user retention throughout the study.

The order in which the two tasks were presented was randomly determined before each session, and the instructions for completing each task were provided before each task begins. The user was required to perform five rounds of each task, with the correctness of answers being validated to ensure the legitimacy of the data and avoid abuse. If the user made a mistake, they were prompted to repeat that round of the task. On completion of both tasks, the touch and sensor data was transmitted to a remote server.

4.4 Task Design

Social media task. The goal of the social media game is to gather touch data by simulating how users tend to use their phones on common vertical scrolling tasks such as browsing a social media feed or looking through a list of news articles. In this task, users were required to scroll through a feed in order to find articles or posts which fit a given description. The articles and corresponding images were gathered from the copyright-free content of NewsUSA [1] and we manually created a non-ambiguous corresponding description for each one of them. Each description was associated with one unique article or post and there were 600 such pairs available in the system. The feed was 20 items long and the correct description-answer pair was randomly chosen and mixed with arbitrary decoys pooled from the rest of the pairs.

Image gallery task. The goal of the image gallery game is to gather touch data by simulating how users tend to use their phones on common horizontal swiping tasks such as browsing a list of photos or application screens. In this task, users were presented with a horizontal list of pictures in which only a single image was visible at any given time. Users were required to count the number of occurrences of a specific object while swiping through the gallery. For instance, the objects could be animals such as dogs and cats or food items such as pizza. All the images were gathered from the open computer vision "Common Objects in Context" (COCO) dataset [21]. There were a total of 200 unique images in the system and each challenge presented 20 images in the gallery while ensuring that between 2 and 6 of them contain the target object. At the end of the round users were required to enter the number of objects they have counted.

The application's source code is available with the rest of the data and code from the project.

4.5 Limitations

As with any remote data collection experiment, the lack of direct experimenter involvement poses challenges. The two actions that could compromise the quality of the dataset are participants completing the study twice or participants asking others to complete some of their sessions. The first case is highly problematic since the user would appear twice in the data under different labels. However, to do so the participant would require two MTurk accounts, two Apple accounts, two physical devices, and the capability to accept and complete the HIT twice before it expires. The second case of participants handing their phones to someone else for some of their sessions is harder to rule out entirely. However, we have reminded participants not to do so at the start of each session and,

the impact of participants disregarding this would be limited to individual sessions.

Lastly, data may have been collected under varying uncontrolled conditions that differ both between users and sessions of the same user. For instance, a user could be sitting or walking, holding the phone, or having it on a table. While this may hinder the overall performance (as it adds variability), it should be considered a more realistic representation of the way a touch-based system will be used in practice.

5 DATASET

In total, we collected data from 470 users amounting to 6,017 unique sessions and 1,166,092 unique swipes. On average, users completed 13 sessions with cumulative distribution function plots for each study duration group shown in Appendix D. The majority of the users that completed the first few sessions continued throughout the whole duration of the experiment. On average, an image gallery task took 1:54 minutes to complete and resulted in the collection of 124 swipes. The social media task took 1:48 minutes to complete and resulted in 79 scrolls using the same method. The average duration of a swipe was 58ms and the average flight time between swipes was 630ms. The demographics of our participants can be found in Appendix D.

6 MACHINE LEARNING PIPELINE

Here we present our data and machine learning pipeline and we describe how we investigate the effect of the pitfalls P2, P3, and P4, which require specific steps. P1 and P5 are analyzed directly by varying the sample size and the aggregation window size, respectively. Our implementation is available online.

Division by phone model. As outlined in Section 4.2, our participants used 9 distinct phone models for data collection. While their hardware and sensors are likely to be very similar, there are differences in their screen size, resolution, and shape. In order to control for the effect of P2, we create distinct data subsets by isolating data collected by each phone model (which we refer to with the phone model name, e.g., XS MAX). We compare the performance on this phone model-specific subsets with the performance computed on the entire dataset containing data from all models, which we refer to as COMBINED.

Preprocessing and feature extraction. As the first step, we aggregate individual touch samples (consisting of X/Y coordinates and touch pressure) within a game into horizontal swipes (image gallery task) and vertical scrolls (social media task). In all future steps, scrolls and swipes are classified separately and independently. In order to avoid including taps, we remove swipes shorter than 3 samples and the ones that do not deviate by more than 5 pixels from the starting point. For each remaining swipe and scroll, we calculate a set of features directly taken from [17]. All positional features are normalized to the screen resolution. We also distinguish between the direction (left/right or up/down) of both swipes and scrolls.

Training data selection. In order to control for the effect of P3, we consider four methods of dividing the target user's data into training and testing sets. In the following, U identifies the set comprising of all users, N_i identifies the number of samples (swipes) belonging to user i, and f_{train} and f_{test} refer to the fraction of samples used for training and testing, respectively.

- RANDOM: we choose training samples for a user out of all the available samples at random, i.e., all sessions are merged, testing uses the remaining samples. This process is repeated independently for each user.
- CONTIGUOUS: we combine all samples of a user and we select the first portion (in chronological order) of samples for training and the remainder for testing.
- DEDICATEDSESSIONS: for a user, we select a subset of their sessions for training and test on the remaining sessions. This ensures that each session is used for either training or testing and that training and testing samples are never drawn from the same session. We investigate selecting sessions both contiguously (in chronological order, with first sessions used for training, later sessions used for testing) and randomly.
- INTRASESSION: for a user, we select a specific session and use the first half of samples for training and the remainder for testing. Only samples from the chosen session are used.

Attacker modeling. To evaluate the effect of P4, we examine two different scenarios, one where attacker samples are included in training data and one where they are not. In both cases, we train a binary model where the user's samples are labeled as positive and multiple other users are combined into a single negative class.

- EXCLUDEATK: For each user we randomly divide the remaining users into two equally-sized sets U_1 and U_2. For training, we select positive class data from the available data from the user and negative class data from U_1. We ensure the two classes are balanced. For testing, we treat all users from U_2 as attackers and classify their samples along with the user's testing samples. This ensures that there is no overlap in the attackers used for training and testing. We use this approach over the leave-one-out method proposed in [12] to avoid overfitting when a separate threshold is chosen for each user-attacker pair.
- INCLUDEATK: We select a user and split the remaining users into U_1 and U_2. We first train and test the system on U_1. This involves training a model for each user i where $N_i * f_{train}$ of the user's samples and $\frac{N_i * f_{train}}{|U_1|}$ of each attacker's samples are used for training and the rest for testing. This ensures that the negative and positive classes are balanced in the training data. This process is then separately repeated with U_2.

Scaling. Following the division of data into training and testing batches along with the inclusion or exclusion of attacker data, we standardize each feature by computing the mean and standard deviation of the training data. The training and testing samples of both the user and the attackers are scaled by subtracting the mean and dividing by the standard deviation of this training data.

Classification. Following scaling, we fit a classifier to our training data for each user. We then classify the samples in the testing set, which gives us a probability for each sample. This probability is in turn used for both sample aggregation and threshold selection.

Sample aggregation. For this optional step, instead of treating samples independently, we group a set of consecutive samples together and take their mean probability estimation, which we use

instead of individual probability estimation for threshold selection and final decision.

Threshold selection. Taking the distance scores for the testing samples (both user and attacker samples), we compute the EER for each user. This is done by finding the distance score threshold where the FAR and FRR are equal. The mean EER for a given system is the average EER across all users.

7 ANALYSIS

To quantify each pitfall's effect on the evaluation performance, we analyze their effect one at a time. Our system implementation is based on one of the seminal papers in the field [17]. We report our results from the SVM classifier as it is the best performing method in the study but also experiment with other classifiers (Random Forest, Neural Network, and k-Nearest Neighbors (kNN)). We discuss classifier differences at the end of this section. When investigating one pitfall, we control the remaining experimental choices estimating a baseline performance as follows: (i) COMBINED, (ii) CONTIGUOUS, (iii) EXCLUDEATK and no sample aggregation. We chose this specific configuration as a default in our experiments for the following reasons. For phone model mixing and training data selection, we chose the most common configurations in Table 1 - COMBINED and CONTIGUOUS respectively. However, we chose EXCLUDEATK as previous work on the topic has already shown the negative effects of using the unrealistic INCLUDATK approach [11]. We do not use an aggregation of samples in our default configuration as it adds another dimension to the data and results, thus making comparison within experiments and previous work more complicated. Unless differently specified, we focus on the effect of pitfalls on the *mean EER*, i.e., for an experiment configuration, we train the system, then use the test set to estimate each user's EER (*per-user EER*) and report the average of those. We also report the mean ROC curve with 95% confidence intervals where appropriate.

The baseline system resulted in a mean EER of 8.4% and a standard deviation of ±5.57. As our goal is to investigate the fundamental effects of evaluation pitfalls, we focus on the most populous left swipe type to limit sources of variability. Details about the per-user EER distribution and effects of swipe direction on performance can be found in Appendix F.

P1: Small sample size

Here we investigate non-trivial effects of user sample size and the effect of the amount of available data per user on the resulting mean EER.

7.0.1 User sample size. Oftentimes it is assumed that the EER of a given authentication method can be reliably estimated by sampling roughly 40 users (the median number of users in Table 1). To investigate this, we randomly sample $n < 470$ users from our dataset and compute the mean EER of the system fit on those n and the standard deviation of each sample's per-user EER distribution. We focus on the standard deviation of the per-user EER distribution as it is a proxy to the evaluation of systematic errors and EER outliers: certain users with high per-user EER are responsible for a larger proportion of the resulting mean EER [12]. The sampling procedure is repeated 1,000 times for each n. We then use n=40 (median user sample size in Table 1) as a reference: we test whether the metrics obtained at n=40 reliably predict the behavior for different n.

Effect on mean EER. The left-hand of Figure 4 reports the difference in behavior between the EER measured empirically for various n and the EER extrapolated from the performance of the n=40 subset. The figure shows that increasing the number of users in the model has a non-negligible effect on the EER: while we obtain EER=9.14% for n=40, increasing the number of users has a large benefit, reaching EER=8.41% for n=400.

Effect on per-user EER standard deviation. The right-hand of Figure 4 reports the difference in behavior between the empirical per-user EER standard deviation for various n and the standard deviation extrapolated from the performance of the n=40 subset. Given the effect described in the previous paragraph, to allow for meaningful comparison we adjust the extrapolated standard deviation to account for the reduction in mean EER (which reduces the per-user EER standard deviation). We do so by adjusting the standard deviation extrapolated at each n with the scaling ration between the empirical mean EER measured at n and the one measured at 40;[2] this moves the two distributions to the same mean EER. Figure 4 (right) shows how for increasing n there is a notable decrement in the per-user EER standard deviation, which is not solely explained by EER mean reduction presented above.

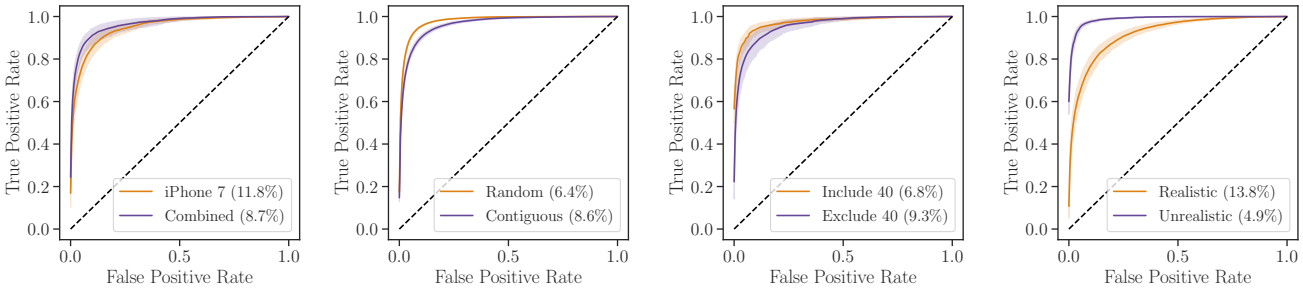
Overall, we find that increasing the user sample size greatly benefits the machine learning model (at least in our general method and SVM), thanks to the added variety of negative samples coming from larger pools of users. Larger sample sizes not only lead to lower and more accurate measurement of underlying EER but also have a regularizing effect on the resulting per-user EER distribution, leading to fewer outliers. This also challenges previous findings regarding the usage of error distribution metrics [12] as user sample sizes also will have an effect on such EER distribution across users.

7.0.2 Number of sessions and swipes. Increasing the amount of data collected per user may lead to differences in performance: (i) across several data collection sessions users may get acclimatized to the task (leading to better stability of the collected swipes) and (ii) larger amount of data per user may generally benefit the performance of the machine learning model. In the following paragraphs, we test both factors separately.

Effect of user acclimatization. We use data from the 68 users who completed the full 31 sessions, given a number of sessions S, we split the data into the earliest collected s sessions (*Early*) and the latest collected s sessions (*Late*). If users gradually get used to the experimental settings (i.e., their behavior exhibits reduced variation), then *Early* sessions will perform worse than *Late* sessions when the user has acclimatized after many repetitions. We apply our authentication pipeline on both early and late sets, making several splits with s ranging from 3 to 15. We report the results in Figure 5, showing no significant difference between the performance of early and late sessions. Therefore the data shows no evidence of task acclimatization leading to changes in performance.

Effect of amount of data per-user. We again use data from the 68 users who completed the full 31 sessions, we consider the effect

[2] given empirical per-user EER standard deviation and EER mean measured at n, σ_n and μ_n, we estimate $\hat{\sigma}_m$ using n=40 as $\hat{\sigma}_m = \frac{\mu_m}{\mu_{40}}\sigma_{40}$.



(a) Phone model mixing **(b) Data selection methods** **(c) Attacker data in training** **(d) Cumulative effects of pitfalls**

Figure 3: ROC Curves for different pitfalls and their cumulative effect. All other parameters are fixed. EER (%) values reported in the legend.

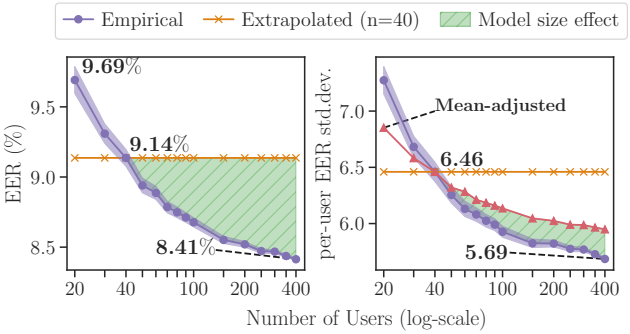


Figure 4: Differences between extrapolated EER using sample size of n=40 and empirical EER measured with various n. Left reports the changes in mean EER, right reports the standard deviation of the per-user EER distribution. Empirical data are computed using 1,000 random n-sized subsamples of our dataset, Extrapolated data are computed generalizing the findings for n=40.

of the increasing amount of data per user by evaluating the system performance as the number of sessions grows. Figure 6 shows the resulting EER for growing number of sessions. We found that no specific trends emerge as the session count varies. We extend the analysis to the remaining users as well by considering the number of swipes per-user rather than the number of sessions. Figure 7 shows the relationship between number of swipes and resulting per-user EER, points are labeled by *Short* or *Long* batch depending on whether the user belonged to either study batch (see Section 4). We found that there is not a clear distinction or trend based on the number of swipes, reinforcing the previous results of Figure 6. Both figures indeed suggest that the number of swipes or sessions does not necessarily affect the performance of our model which contradicts hypothesis (ii). While long-term studies are necessary to investigate the stability of the biometric, the availability of long-term data does not affect EER in a significant way.

P2: Phone model mixing

In this section, we compare the system performance on data belonging to individual phone models and when merging together data from various phone models (COMBINED). We then explore this concept further by measuring how accurately we can predict the phone model a swipe originated from.

Effect of combining phone models. As evidenced in the previous Section 7, increasing n leads to an EER reduction (see Figure 4).

Table 2: Model performance when training and testing with the same phone model or when mixing phone models (COMBINED). COMBINED results in overestimation of performance even when subsampling to the number of users present in each specific phone model.

Model	Users (n)	Mean EER (CI 95%)	COMBINED EER (CI 95%)	P-value
6S	70	12.3% (±2.46)	8.8% (±2.04)	**.032**
6S PLUS	19	14.2% (±6.28)	9.9% (±4.00)	.233
7	73	11.8% (±1.60)	8.7% (±1.17)	**.002**
7 PLUS	50	11.6% (±2.19)	9.1% (±1.81)	.082
8	68	12.4% (±1.84)	8.8% (±1.14)	**.001**
8 PLUS	55	12.7% (±2.32)	9.0% (±1.94)	**.014**
X	71	13.1% (±2.03)	8.8% (±1.68)	**.002**
XS	34	13.6% (±3.01)	9.1% (±2.01)	**.014**
XS MAX	30	12.9% (±4.01)	9.3% (±2.66)	.135

To account for this, we compare each single-phone subset to a COMBINED subsample from all phone models, with an equal number of users as for each respective phone model. Table 2 presents the results for COMBINED dataset and single-phone model subsets. The table shows that the COMBINED approach leads to an overestimation of performance. We observed a decrease in EER for each of the phone models. Furthermore, we performed a t-test and found that the EER difference between a single phone model and a subsample is statistically significant ($P < .05$) except for 6S PLUS, 7 PLUS and XS MAX. Figure 3a shows the complete ROC curves for the iPhone 7 model (which includes the most number of users in our dataset) and its respective COMBINED model. The overestimation of performance is present throughout the whole of the ROC curves apart from extreme TPR and FPR values. The ROC curves for the other phone models can be found in Appendix A.

Phone model identifiability. We create a phone model classifier whose aim is to identify the iPhone model of a given swipe. We merge all the available data and label each swipe with its originating phone model; data is then divided into 80/20 train-test splits. The data is balanced such that each phone model had an equal number of swipes in the training split. We make sure that users which were used in training were not considered in testing and vice versa (to avoid biasing the prediction with the users' identities). We fit an SVM classifier with the data. We perform this experiment once using

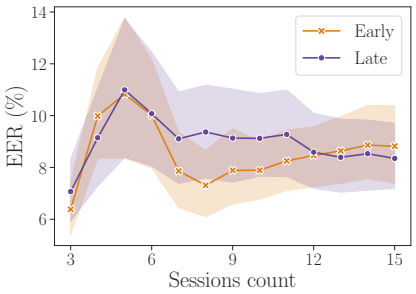


Figure 5: EERs of Early and Late session subsets for users with 31 completed sessions (n=68). No significant difference is found between the subsets, suggesting user task familiarization does not affect behavior.

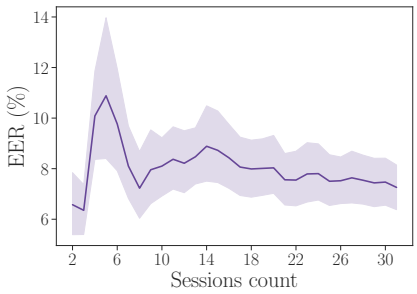


Figure 6: EERs when considering an increasing number of sessions for users with 31 completed sessions (n=68). The shaded areas report 95% confidence intervals.

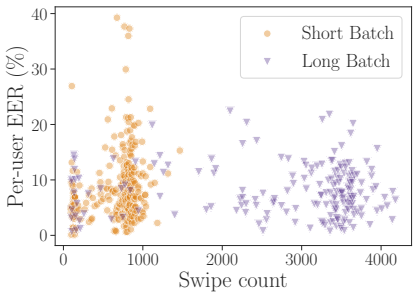


Figure 7: Relationship between per-user EER and number of swipes available for each user. Short and Long batch labels mark the studies users belong to (see Section 4).

all 9 phone models and again only with the 6s, 7 and 8 models as these three have equal screen sizes, resolutions, and pixel densities. The classifier achieves 44% accuracy where a random baseline model would yield 11.1%. When considering only 6s, 7 and 8, we achieved an accuracy of 49% compared to a baseline of 33.3%. A complete confusion matrix for the classification of the experiments including all nine phone models can be found in Appendix G. This shows that differences in the properties of the devices are reflected in the identification outcome, i.e., swipes belonging to similar phone models tend to be more similar.

These results indicate that it is undesirable to mix different phone models in data collection and analysis for touch-based authentication. Furthermore, it is irrelevant whether the mixed models have similar screen sizes, dimensions, or display pixel densities. The practice of mixing phone models can lead to an artificial increase of performance between 2.5% and 4.5% EER.

P3: Non-contiguous training data selection

We compared the classification performance of our model under the conditions described in Section 6: (i) RANDOM, (ii) CONTIGUOUS, (iii) DEDICATEDSESSIONS and (iv) INTRASESSION. For a fair comparison, we only used data from the 409 users which have completed 2 or more sessions as this is a prerequisite for the DEDICATEDSESSIONS modality. We present our findings in Table 3. As expected the INTRASESSION method yielded the best performance as users have a more stable interaction pattern during a single session than through time [17]. The fact that the model performed well in this category is hopeful, but in practice, users carry out many sessions throughout time and the INTRASESSION result should not be considered an accurate metric for touch-based authentication systems. Mixing and randomizing samples from all sessions (RANDOM approach) provided a similar effect as the model learns on information about users' interactions throughout all sessions. CONTIGUOUS training also allows the model to learn from an overlapping session, which yields better performance. The DEDICATEDSESSIONS scenario is the most realistic one for a touch authentication system as it relies on self-contained training sessions - as they will be performed in a deployed system.

We found that results between all of the methods vary considerably and performance seems to be overestimated compared to the realistic DEDICATEDSESSIONS approach. An unrealistic training data selection can lead to an increase in performance of 3.8% EER when using a RANDOM approach compared to the DEDICATEDSESSIONS approach. The complete ROC curves resulting from this experiment are available in Figure 3c. The ROC curve results are mostly consistent with the EER reported in Table 3 apart from RANDOM and INTRASESSION curves where RANDOM selection has a higher TPR above 0.08% FPR.

Table 3: Model performance for common training data selection approaches. Random selection results in overestimated performance.

Data Selection Method	Mean EER (%)	CI (95%)
RANDOM	6.4	± 0.28
CONTIGUOUS	8.6	± 0.55
DEDICATEDSESSIONS (Contiguous)	10.1	± 0.70
DEDICATEDSESSIONS (Random)	10.2	± 0.68
INTRASESSION	5.6	± 0.25

P4: Attacker data in training

We compared different attack modeling choices as described in Section 6: (i) EXCLUDEATK and (ii) INCLUDEATK. To do so, we randomly subsampled n users from our dataset at various n, for each n we apply our pipeline and compute the resulting EER for the two approaches. This procedure is repeated 10 times, Figure 8 and Figure 9 illustrate the results. We find that INCLUDEATK results in consistently lower mean EER when compared to EXCLUDEATK, see Figure 8. However, Figure 9 shows how the EER difference between the two approaches decreases exponentially as the number of users (n) increases. This is expected as the fewer users are considered the more the presence of attacker data impacts the classifier (e.g., 10% of negative training data for n=11 users, <1% of negative training data for $n > 101$ users). This diminishing return also explains why in INCLUDEATK the EER increases when more users are included, despite the expectation that more data might result in better performance. Figure 9 shows that at n=40, the EER difference between the two approaches is 2.55%. As pointed out in Table 1, 80% of our reported studies falls into P4, meaning that these might not present performance metrics appropriate for the specified threat model. Overall,

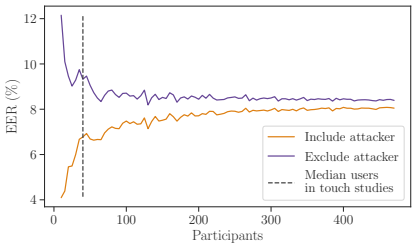


Figure 8: Resulting mean EER when using IncludeAtk and ExcludeAtk attacker modeling approaches. We report the mean of the EER across 10 random subsampling repetitions. A large EER difference is observed when considering a small number of users.

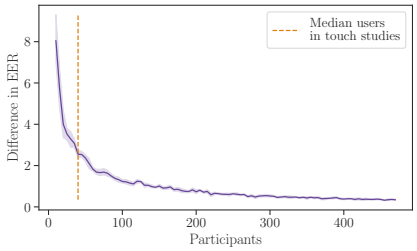


Figure 9: Absolute EER difference between IncludeAtk and ExcludeAtk attacker modeling approaches. For each number of users, the shaded areas report 95% confidence intervals on the mean difference from 10 random subsampling repetitions.

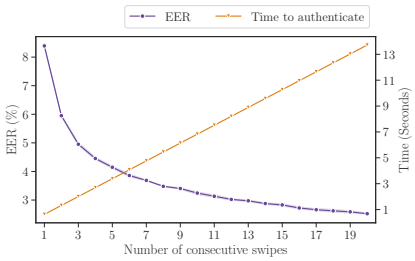


Figure 10: Performance of an aggregation model which selects the mean distance scores of a number of consecutive swipes before calculating EER. The shaded areas report 95% confidence intervals on the mean EER from 10 repetitions.

depending on the user sample size considered, IncludeAtk can lead to an artificial performance gain of between 0.3% and 6.9%. Figure 3c shows the ROC curves of IncludeAtk and ExcludeAtk models for 40 users (the average number of users from Table 1). The ROC curves for 20, 100, 200, 300 and 400 users are also available in Appendix B.

P5: Aggregation window size

When reporting their results, many studies [17, 19, 20, 32, 44] consider the performance of a group of consecutive swipes instead of a single one as we have done so far. Figure 10 shows the performance of our pipeline when we use an aggregation of consecutive swipes as described in Section 6. The procedure was repeated 10 times and shaded areas show the 95% confidence interval across the ten repetitions. As expected, increasing the aggregation window size leads to lower EERs: an EER of 8.2% obtained on single swipes drops more than a quarter (5.9%) when aggregating two swipes, and drops to less than 3% at 12 swipes. Touch-based authentication studies should be clear when and how they use such aggregations as they evidently have an impact on performance. It should also be noted that each swipe action takes time to perform which can leave a system at risk. For instance, our dataset suggests that on the tasks considered, performing 20 swipes would take 14 seconds during which the system would be vulnerable. Therefore a balance between usability and security should be sought.

Cumulative effects of evaluation choices

In this subsection, we quantify the difference between *realistic* (pitfall-free) and *unrealistic* (with all pitfalls) evaluation choices for touch authentication systems. We repeated the following two procedures 100 times and report the mean of all runs and the confidence interval at 95%. In the unrealistic methods experiment, we combined phone models (Combined), included the attacker into the training data (IncludeAtk), used the Random data selection method and each round randomly subsampled our dataset to the median of n=40 participants taken from Table 1 (to even out the effect of P1). This resulted in a 4.9% EER with a confidence interval of ±0.09. In the realistic method experiment, again we selected n=40 users from the most commonly used iPhone 7 phone model, used ExcludeAtk and the DedicatedSessions training data selection.

Table 4: Impact of pitfalls on different classifiers. The table presents the percentage-point difference in EER between using realistic and unrealistic evaluation methods.

Pitfall	SVM	Random Forest	Neural Network	kNN
P1 400 users vs 40 users	0.72	0.28	0.87	1.25
P2 iPhone 7 vs Combined	4.08	4.53	2.40	3.29
P3 Contiguous vs Randomized	2.27	2.62	2.06	2.35
P4 Exclude vs Include	2.55	2.69	3.41	3.96
Cumulative Impact	8.89	10.36	8.99	9.79

Each round we randomly select which users are selected as attackers. This approach resulted in a much worse EER of 13.8% with a confidence interval of ±0.14. Figure 3d illustrates the overestimation of performance throughout the ROC curves of these experiments. The results clearly illustrate that flawed methods have strong effects on the resulting performance and can lead to an artificial boost to performance by 8.9% EER.

Effects of classifiers on evaluation choices

In this subsection, we quantify the impact of pitfalls on performance on four of the most widely used machine learning algorithms in the field. Implementation details for each individual classifier can be found in Appendix C. The results of our experiments are presented in Table 4. All of the examined pitfalls introduce an overestimation of performance regardless of the classifier chosen. However, there are differences in individual performance across chosen classifiers. For instance, the kNN classifier relies heavily on individual swipes similar to the target one, hence the impact of including the attacker data into training is much more pronounced. These results suggest that the pitfalls apply to a wide range of touch dynamics system implementations.

8 BEST PRACTICES

In order to facilitate better comparison between future studies and achieve unbiased performance evaluation, we propose a standard set of practices to follow when evaluating touch-based authentication systems, derived from our set of common evaluation pitfalls.

P1: Small sample size. While it is hard to advocate for a specific minimum number of users to be required by a study, we recommend researchers to be aware of the effects of user sample sizes in pipelines similar to the one analyzed in this paper. Based on the findings in Section 7, we found that increasing sample size has two important effects: it reduces the resulting mean EER and smooths the variance of the per-user EER distribution. It is advisable that an analysis of the effect of sample size is included in new studies, and that results for a sample size of n=40 are also reported (when applicable). This best practice must be accounted for at the study design phase, to ensure enough data is initially collected.

P2: Phone model mixing. A single phone model should be used to train and test a proposed system. While this might not always be the final use case (e.g., in other scenarios, one might want to test the generalization performance of a device-specific classifier on a different device), this avoids the bias introduced by data collected on a specific phone model. Isolating data belonging to different phone models when training will produce more accurate performance measurements. Care must be taken in data collection to ensure there are enough samples for each phone model that will be studied.

P3: Non-contiguous training data selection. Randomized swipes selection should not be used to separate training and testing data. Test data must always have been collected at a time after the training data was collected, to mimic real-world usage, and to account for behavior drift. For comparison between works, only an initial training phase (enrollment) should be included, as training updates increase the difficulty of comparing figures. Ideally, at least two sessions should be used to collect training and test data, as the bulk of real-world usage occurs with a time interval between enrollment and authentication.

P4: Attacker data in training. Studies should always exclude the attacker from the training set, as one shall never assume they have information about the attacker in a deployed system. In particular, care should be taken so that any attacker of a model was not included as a negative example when training the model. Excluding the attacker is particularly important with studies with a limited number of users, where the effect of such an attacker modeling approach greatly affects the resulting performance.

P5: Aggregation window size. Using aggregation of consecutive swipes is beneficial to performance, particularly when using the mean of their distances to the decision boundary as shown in Fig 10. However, researchers should report the performance of a single swipe model in order to ensure comparability with other studies, as well as other reasonable numbers of swipes that other similar papers have proposed. Furthermore, information about the flight time between swipes and their duration should also be shared, as these directly relate to the time the system is vulnerable to an attacker.

P6: Dataset and code availability. Historically, in this field, it has been rare for authors to share their data – see Table 1 – and none of the studies examined in the related work share their analysis code. This leads to uncertainty when reproducing results, in fact, for some studies, it was unclear from the paper alone whether the study made certain choices regarding the experiments (e.g., we could not clearly define whether 30% of studies fell into P3). The code and datasets of touch authentication studies should be made freely available. This ensures that results can be reproduced by others, and reduces barriers to entry of those wishing to build upon existing work.

Generality of results. Although this paper focuses on touch-based authentication, we believe these best practices apply in similar ways to other types of biometric systems such as facial recognition and keystroke authentication. In particular, non-contiguous training data selection (P3), and inclusion of attacker data in training (P4) are fundamentally flawed and should be avoided in all biometric system evaluations. However, the effect of mixing similar devices (P2) may vary across different modalities. Similarly, the sample size implications (P1) might differ in other systems from what we found in our experimentation. Nevertheless, these points should be examined with caution by the relevant literature.

Further work is required to examine to what extent these pitfalls are prevalent in the study of other biometric authentication systems.

9 CONCLUSION

In this work, we explored the impacts of evaluation choices on touch-based authentication methods. We investigated performance differences in approaches related both to data gathering and choices in the way classifiers are trained with a certain data split. For the purpose of this study, we collected a large open-source dataset for touch-based mobile authentication consisting of 470 users, which we made publicly available. We confirmed large variations in performance based on phone model mixing (up to 5.8% EER), training data selection (up to 3.8% EER), user sample size (up to 4% EER), and attacker modeling (up to 6.9% EER). Finally, combining all evaluation pitfalls results in overestimation of performance by 8.9% EER. The results are largely similar regardless of the chosen classifier. We also note that, aside from some extreme threshold settings, these effects are observable throughout the ROC curve. Based on these findings, we proposed a set of good practices to be considered in order to enable accurate reporting of results and to allow comparability across studies.

REFERENCES

[1] [n. d.]. NewsUSA: Copyright Free Content. https://www.copyrightfreecontent.com/category/newsusa/. Accessed: 15 November 2021.

[2] Alejandro Acien, Aythami Morales, Julian Fierrez, Ruben Vera-Rodriguez, and Oscar Delgado-Mohatar. 2021. BeCAPTCHA: Behavioral bot detection using touchscreen and mobile sensors benchmarked on HuMIdb. *Engineering Applications of Artificial Intelligence* 98 (2021), 104058. https://doi.org/10.1016/j.engappai.2020.104058

[3] Jamil Ahmad, Muhammad Sajjad, Zahoor Jan, Irfan Mehmood, Seungmin Rho, and Sung Wook Baik. 2017. Analysis of interaction trace maps for active authentication on smart devices. *Multimedia Tools and Applications* 76, 3 (01 Feb 2017), 4069–4087. https://doi.org/10.1007/s11042-016-3450-y

[4] Shatha J. Alghamdi and Lamiaa A. Elrefaei. 2018. Dynamic Authentication of Smartphone Users Based on Touchscreen Gestures. *Arabian Journal for Science and Engineering* 43, 2 (01 Feb 2018), 789–810. https://doi.org/10.1007/s13369-017-2758-x

[5] Margit Antal, Zsolt Bokor, and László Zsolt Szabó. 2015. Information revealed from scrolling interactions on mobile devices. *Pattern Recognition Letters* 56 (2015), 7 – 13. https://doi.org/10.1016/j.patrec.2015.01.011

[6] Margit Antal and László Zsolt Szabó. 2016. Biometric Authentication Based on Touchscreen Swipe Patterns. *Procedia Technology* 22 (2016), 862 – 869. https://doi.org/10.1016/j.protcy.2016.01.061 9th International Conference Interdisciplinarity in Engineering, INTER-ENG 2015, 8-9 October 2015, Tirgu Mures, Romania.

[7] Margit Antal and László Zsolt Szabó. 2016. Biometric Authentication Based on Touchscreen Swipe Patterns. *Procedia Technology* 22 (2016), 862 – 869. https://doi.

org/10.1016/j.protcy.2016.01.061 9th International Conference Interdisciplinarity in Engineering, INTER-ENG 2015, 8-9 October 2015, Tirgu Mures, Romania.

[8] Cheng Bo, Lan Zhang, Xiang-Yang Li, Qiuyuan Huang, and Yu Wang. 2013. SilentSense: Silent User Identification via Touch and Movement Behavioral Biometrics. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking* (Miami, Florida, USA) *(MobiCom '13)*. Association for Computing Machinery, New York, NY, USA, 187–190. https://doi.org/10.1145/2500423.2504572

[9] Daniel Buschek, Alexander De Luca, and Florian Alt. 2016. Evaluating the Influence of Targets and Hand Postures on Touch-Based Behavioural Biometrics. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1349–1361. https://doi.org/10.1145/2858036.2858165

[10] D. Deb, A. Ross, A. K. Jain, K. Prakah-Asante, and K. V. Prasad. 2019. Actions Speak Louder Than (Pass)words: Passive Authentication of Smartphone* Users via Deep Temporal Features. In *2019 International Conference on Biometrics (ICB)*.

[11] Simon Eberz, Kasper B. Rasmussen, Vincent Lenders, and Ivan Martinovic. 2017. Evaluating Behavioral Biometrics for Continuous Authentication: Challenges and Metrics. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (Abu Dhabi, United Arab Emirates) *(ASIA CCS '17)*. ACM, New York, NY, USA, 386–399. https://doi.org/10.1145/3052973.3053032

[12] Simon Eberz, Kasper B Rasmussen, Vincent Lenders, and Ivan Martinovic. 2017. Evaluating behavioral biometrics for continuous authentication: Challenges and metrics. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. 386–399.

[13] T. Feng, Z. Liu, K. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen. 2012. Continuous mobile authentication using touchscreen gestures. In *2012 IEEE Conference on Technologies for Homeland Security (HST)*. 451–456. https://doi.org/10.1109/THS.2012.6459891

[14] T. Feng, Z. Liu, K. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen. 2012. Continuous mobile authentication using touchscreen gestures. In *2012 IEEE Conference on Technologies for Homeland Security (HST)*. 451–456. https://doi.org/10.1109/THS.2012.6459891

[15] Tao Feng, Jun Yang, Zhixian Yan, Emmanuel Munguia Tapia, and Weidong Shi. 2014. TIPS: Context-Aware Implicit User Identification Using Touch Screen in Uncontrolled Environments. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications* (Santa Barbara, California) *(HotMobile '14)*. Association for Computing Machinery, New York, NY, USA, Article 9, 6 pages. https://doi.org/10.1145/2565585.2565592

[16] J. Fierrez, A. Pozo, M. Martinez-Diaz, J. Galbally, and A. Morales. 2018. Benchmarking Touchscreen Biometrics for Mobile Authentication. *IEEE Transactions on Information Forensics and Security* 13, 11 (2018), 2720–2733.

[17] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. 2013. Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. *IEEE Transactions on Information Forensics and Security* 8, 1 (2013), 136–148.

[18] Hassan Khan and Urs Hengartner. 2014. Towards Application-Centric Implicit Authentication on Smartphones. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications* (Santa Barbara, California) *(HotMobile '14)*. Association for Computing Machinery, New York, NY, USA, Article 10, 6 pages. https://doi.org/10.1145/2565585.2565590

[19] R. Kumar, V. V. Phoha, and A. Serwadda. 2016. Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 1–8. https://doi.org/10.1109/BTAS.2016.7791164

[20] Lingjun Li, Xinxin Zhao, and Guoliang Xue. 2013. Unobservable Re-authentication for Smartphones. In *20th Annual Network and Distributed System Security Symposium, NDSS 2013, San Diego, California, USA, February 24-27, 2013*. The Internet Society.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 740–755.

[22] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa. 2016. Active user authentication for smartphones: A challenge data set and benchmark results. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 1–8.

[23] U. Mahbub, S. Sarkar, V. M. Patel, and R. Chellappa. 2016. Active user authentication for smartphones: A challenge data set and benchmark results. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 1–8. https://doi.org/10.1109/BTAS.2016.7791155

[24] Weizhi Meng, Yu Wang, Duncan S. Wong, Sheng Wen, and Yang Xiang. 2018. TouchWB: Touch behavioral user authentication based on web browsing on smartphones. *Journal of Network and Computer Applications* 117 (2018), 1–9. https://doi.org/10.1016/j.jnca.2018.05.010

[25] Yuxin Meng, Duncan S. Wong, Roman Schlegel, and Lam-for Kwok. 2013. Touch Gestures Based Biometric Authentication Scheme for Touchscreen Mobile Phones. In *Information Security and Cryptology*, Mirosław Kutyłowski and Moti Yung (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 331–350.

[26] Rahul Murmuria, Angelos Stavrou, Daniel Barbará, and Dan Fleck. 2015. Continuous Authentication on Mobile Devices Using Power Consumption, Touch Gestures and Physical Movement of Users. In *Research in Attacks, Intrusions, and Defenses*, Herbert Bos, Fabian Monrose, and Gregory Blanc (Eds.). Springer International Publishing, Cham, 405–424.

[27] Rahul Murmuria, Angelos Stavrou, Daniel Barbará, and Dan Fleck. 2015. Continuous Authentication on Mobile Devices Using Power Consumption, Touch Gestures and Physical Movement of Users. In *Research in Attacks, Intrusions, and Defenses*, Herbert Bos, Fabian Monrose, and Gregory Blanc (Eds.). Springer International Publishing, Cham, 405–424.

[28] J. Nader, A. Alsadoon, P.W.C. Prasad, A.K. Singh, and A. Elchouemi. 2015. Designing Touch-Based Hybrid Authentication Method for Smartphones. *Procedia Computer Science* 70 (2015), 198 – 204. https://doi.org/10.1016/j.procs.2015.10.072 Proceedings of the 4th International Conference on Eco-friendly Computing and Communication Systems.

[29] Michail D. Papamichail, Kyriakos C. Chatzidimitriou, Thomas Karanikiotis, Napoleon-Christos I. Oikonomou, Andreas L. Symeonidis, and Sashi K. Saripalle. 2019. BrainRun: A Behavioral Biometrics Dataset towards Continuous Implicit Authentication. *Data* 4, 2 (2019). https://doi.org/10.3390/data4020060

[30] Rodrigo Rocha, Davide Carneiro, and Paulo Novais. 2020. Continuous authentication with a focus on explainability. *Neurocomputing* (2020). https://doi.org/10.1016/j.neucom.2020.02.122

[31] Premkumar Saravanan, Samuel Clarke, Duen Horng (Polo) Chau, and Hongyuan Zha. 2014. LatentGesture: Active User Authentication through Background Touch Analysis. In *Proceedings of the Second International Symposium of Chinese CHI* (Toronto, Ontario, Canada) *(Chinese CHI '14)*. Association for Computing Machinery, New York, NY, USA, 110–113. https://doi.org/10.1145/2592235.2592252

[32] A. Serwadda, V. V. Phoha, and Z. Wang. 2013. Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. 1–8. https://doi.org/10.1109/BTAS.2013.6712758

[33] C. Shen, Y. Zhang, X. Guan, and R. A. Maxion. 2016. Performance Analysis of Touch-Interaction Behavior for Active Smartphone Authentication. *IEEE Transactions on Information Forensics and Security* 11, 3 (2016), 498–513. https://doi.org/10.1109/TIFS.2015.2503258

[34] Shridatt Sugrim, Can Liu, Meghan McLean, and Janne Lindqvist. [n. d.]. Robust Performance Metrics for Authentication Systems. *Network and Distributed Systems Security (NDSS) Symposium 2019* ([n. d.]).

[35] Zahid Syed, Jordan Helmick, Sean Banerjee, and Bojan Cukic. 2019. Touch gesture-based authentication on mobile devices: The effects of user posture, device size, configuration, and inter-session variability. *Journal of Systems and Software* 149 (2019), 158–173. https://doi.org/10.1016/j.jss.2018.11.017

[36] M. Temper, S. Tjoa, and M. Kaiser. 2015. Touch to Authenticate — Continuous Biometric Authentication on Mobile Devices. In *2015 1st International Conference on Software Security and Assurance (ICSSA)*. 30–35. https://doi.org/10.1109/ICSSA.2015.016

[37] Xiao Wang, Tong Yu, Ole Mengshoel, and Patrick Tague. 2017. Towards Continuous and Passive Authentication across Mobile Devices: An Empirical Study. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (Boston, Massachusetts) *(WiSec '17)*. Association for Computing Machinery, New York, NY, USA, 35–45. https://doi.org/10.1145/3098243.3098244

[38] Yuji Watanabe and Liu Kun. 2017. Long-Term Influence of User Identification Based on Touch Operation on Smart Phone. *Procedia Comput. Sci.* 112, C (Sept. 2017), 2529–2536. https://doi.org/10.1016/j.procs.2017.08.196

[39] C. Wu, K. He, J. Chen, and R. Du. 2019. ICAuth: Implicit and Continuous Authentication When the Screen Is Awake. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 1–6.

[40] Hui Xu, Yangfan Zhou, and Michael R. Lyu. 2014. Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones. In *Proceedings of the Tenth USENIX Conference on Usable Privacy and Security* (Menlo Park, CA) *(SOUPS '14)*. USENIX Association, USA, 187–198.

[41] Yafang Yang, Bin Guo, Zhu Wang, Mingyang Li, Zhiwen Yu, and Xingshe Zhou. 2019. BehaveSense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics. *Ad Hoc Networks* 84 (2019), 9–18. https://doi.org/10.1016/j.adhoc.2018.09.015

[42] V. Zaliva, W. Melicher, S. Saha, and J. Zhang. 2015. Passive user identification using sequential analysis of proximity information in touchscreen usage patterns. In *2015 Eighth International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*. 161–166. https://doi.org/10.1109/ICMU.2015.7061060

[43] H. Zhang, V. M. Patel, M. Fathy, and R. Chellappa. 2015. Touch Gesture-Based Active User Authentication Using Dictionaries. In *2015 IEEE Winter Conference on Applications of Computer Vision*. 207–214. https://doi.org/10.1109/WACV.2015.35

[44] X. Zhao, T. Feng, W. Shi, and I. A. Kakadiaris. 2014. Mobile User Authentication Using Statistical Touch Dynamics Images. *IEEE Transactions on Information Forensics and Security* 9, 11 (2014), 1780–1789. https://doi.org/10.1109/TIFS.2014.2350916

[45] N. Zheng, K. Bai, H. Huang, and H. Wang. 2014. You Are How You Touch: User Verification on Smartphones via Tapping Behaviors. In *2014 IEEE 22nd International Conference on Network Protocols*. 221–232. https://doi.org/10.1109/ICNP.2014.43

A PHONE MODEL MIXING ROC CURVES

Figure 11 shows the ROC curves for individual phone models compared to mixing them. We found that our results are largely consistent throughout the length of the ROC curve.

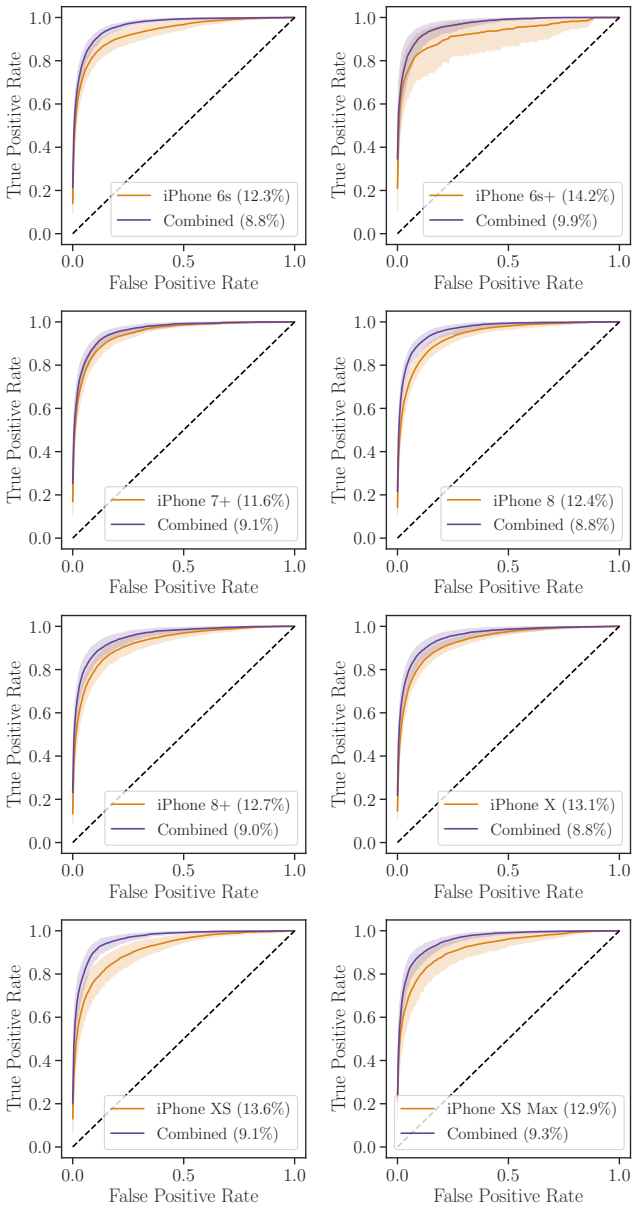


Figure 11: ROC Curves for individual phone models compared to COMBINED models which use the same number of users but merge multiple phone models.

B ATTACKER DATA IN TRAINING ROC CURVE

Figure 12 shows the ROC curves for models which include or exclude attacker from the training data. We present our results for samples of 20, 40, 100, 200, 300, and 400 users. We found that our results are largely consistent throughout the length of the ROC curve.

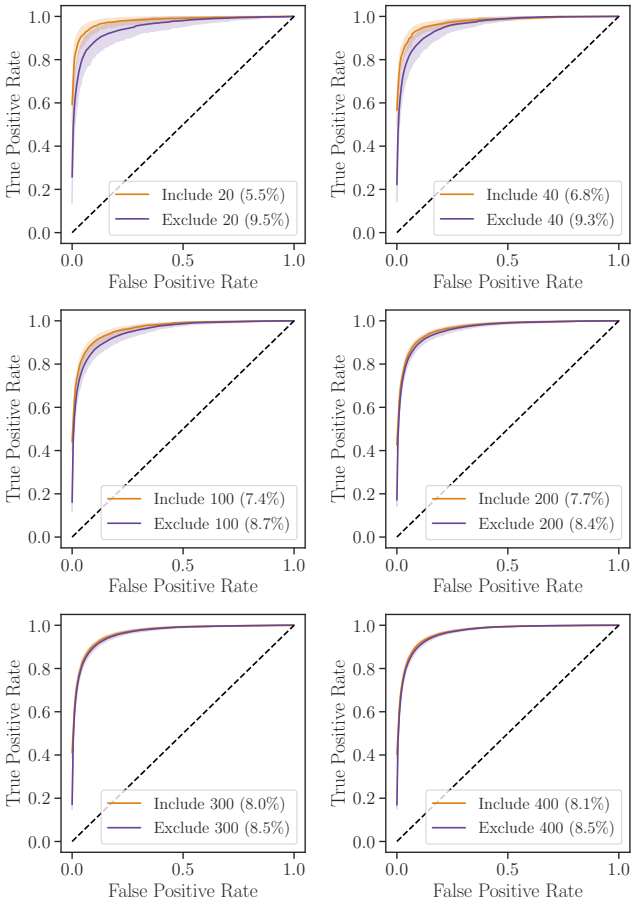


Figure 12: ROC Curves for including or excluding attacker data into the training set of a model at different sample sizes.

C CLASSIFIER IMPLEMENTATION DETAILS

We use the SVM, Random Forest, and kNN classifier implementation of the widely used machine learning library `scikit-learn`. The former two classifiers use the default parameters of the framework and we choose n=18 for the kNN classifier based on preliminary experimentation. Our neural network implementation uses the machine learning libraries Tensorflow and Keras. The feed-forward network consists of 3 hidden layers of sizes 30,30 and 15 with batch normalization and dropout layer (0.3) between them. The optimizer is Adam and the activation function is ReLU. Similarly, we chose the set parameters based on preliminary experimentation.

D DATASET DEMOGRAPHICS

The use of remote collection through the MTurk platform organically resulted in a relatively balanced dataset in terms of age, gender, handedness, and iPhone model. The gender distribution of all users was 47% females (229), 51% males (252), and 1% other (5). Only 14% (67) of the participants reported being left-handed which is roughly comparable to 10% in the general population. The age distribution of participants is shown in Figure 13.

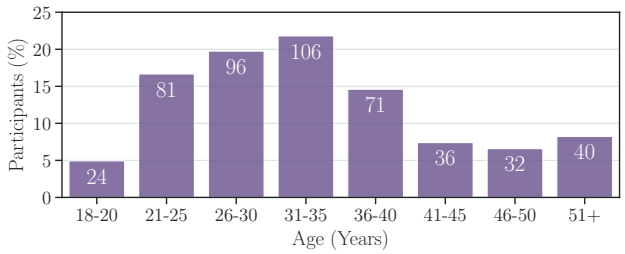


Figure 13: Age of participants in the experiment. Remote collection through Amazon Mechanical Turk allows for more diverse participants compared to traditional university lab studies.

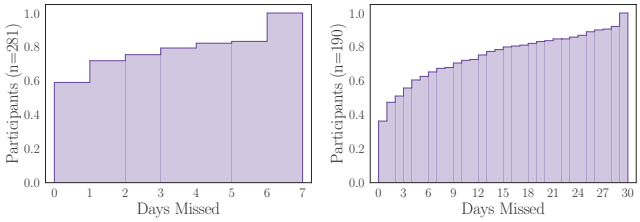


Figure 14: Cumulative distribution function (CDF) of participation retention for seven-day (left) and 31-day (right) user batches.

E DEVICES USED FOR DATA COLLECTION

Table 5 presents the 9 iPhone models we used for our experiments in order of their release dates.

Table 5: Specification sheet details for iPhone models used.

Model	Screen size (in)	Resolution	Pixel density (ppi)
6S	4.7	1334x750	326
6S Plus	5.5	1920x1080	401
7	4.7	1334x750	326
7 Plus	5.5	1920x1080	401
8	4.7	1334x750	326
8 Plus	5.5	1920x1080	401
X	5.8	2436x1125	458
XS	5.8	2436x1125	458
XS Max	6.5	2688x1242	458

F GENERAL SYSTEM RESULTS

The per-user EER distribution of our baseline model is shown in Figure 15. We repeat our baseline model for each swipe direction and report the result in Table 6 together with the amount of data available for each swipe direction. Down and right swipes are underrepresented as these interactions are performed rarely in our application, leading to much higher mean EERs of up to 19% and 16.2%, respectively.

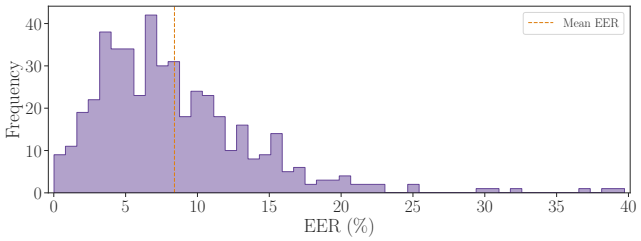


Figure 15: Per-user EER distribution using all users in our dataset (n=470). The performance results in a positively skewed distribution.

Table 6: Model performance for varying swipe directions.

Direction	Count	Mean EER (%)	Std. Dev.
Scroll Up	376,236	10.1	7.2
Scroll Down	45,737	19.0	11.9
Swipe Left	718,036	8.4	5.6
Swipe Right	26,083	16.2	10.5

G PHONE MODEL IDENTIFIABILITY

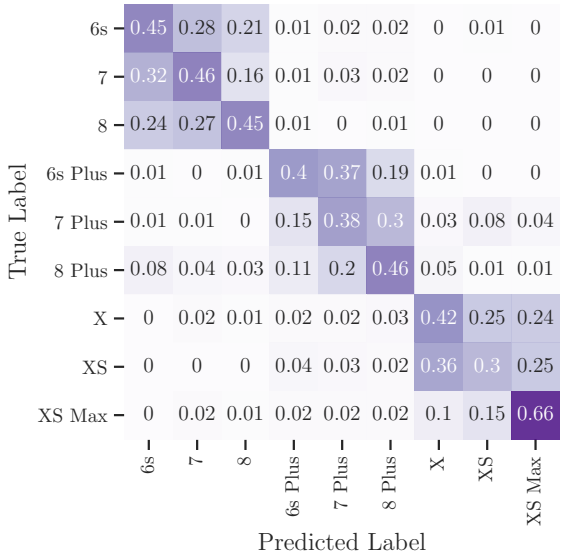


Figure 16: Confusion matrices of phone model prediction for the nine iPhone models in our study. The model prediction errors are concentrated in phones with similar dimensions and resolutions.