Measuring How We Play: Authenticating Users with Touchscreen Gameplay

Jonathan Voris

New York Institute of Technology, New York NY 10023, USA
jvoris@nyit.edu

Abstract. Mobile devices are being used to access and store an ever-increasing amount of sensitive data. Due to their compact form factor, mobile devices can be easily lost or stolen. Yet users frequently choose not to enable authentication or select authentication methods which are insufficient to protect their devices, placing user information at risk. In this paper, we propose the use of a behavioral biometric based approach to authentication that functions by modeling the manner in which users interact with mobile games, which are one of the most popular uses of mobile devices. We conducted an IRB approved study in which 30 participants were asked to play three popular Android games as well as utilize a mobile touchscreen without any gameplay prompting. We extracted features from users' touchscreen activity during these interactions, then applied a Support Vector Machine to classify users based on patterns which emerged from their usage during the game. Our results indicate that using gameplay as a behavioral biometric is an effective means of authenticating users to their mobile devices, but care must be taken to select a game which encourages users to make frequent distinctive gestures.

Key words: active authentication, behavioral biometrics, games for security, gamification, machine learning, mobile authentication, svm, useful games

1 Introduction

Smartphone penetration rates have grown dramatically worldwide over the past decade. People have become accustomed to using their mobile devices to perform a greater variety of tasks, which has caused these devices to store and access an increasingly large amount of sensitive data. In many cases the data accessible via a mobile device is of greater value than the physical device itself. A recent study revealed that 50% of phone theft victims would pay $500 and 33% would pay $1,000 to retrieve their stolen devices; moreover, to regain their handset, 68% of victims would put themselves in danger [11].

Strong authentication is critically important to the process of securing the sensitive data stored on mobile devices. Unfortunately, many people underestimate the importance of the security of their devices. According to Consumer Reports, 36% of American smartphone owners use simple 4 digit numeric passcodes to protect their devices, while 34% choose not to enable any authentication mechanism at all. [18]. Although multiple factors inform users' security decisions, one of the reasons for this missing layer of security is that many mobile authentication methods fail to take usability into consideration.

In an effort to provide users with more usable authentication to their mobile devices, we consider an alternative authentication mechanism which utilizes the process of playing a touchscreen game on a mobile device. Specifically, in order to unlock a device, users are

required to play a particular game for which the proposed system has learned the user's behavior by constructing a model of inherent gameplay characteristics. The motivation behind the selection of games as a potential avenue for authentication is that the act of playing games is one of the most popular activities performed on mobile devices. As of 2016, 57% of mobile users have games installed on their phones [12]. Further, consumers spend 1.15 billion hours each month playing games, ranking them as the second most popular mobile activity following social media [12]. We thus explore applications of gameplay to the security task of authentication because of the natural usability benefits they confer as well as the fact that mobile device owners are already acclimated to playing them.

Note that unlike traditional authentication methods, such as passwords, our game-centric authentication solution is not knowledge-based; that is, users do not need to remember a pre-established secret in order to gain access to their mobile system. Instead, users are authenticated based on whether or not the patterns which emerge from how they interact with a game match or deviate from a model of how the legitimate device owner has played in the past. This provides several advantages over secret-based approaches to authentication, primarily that behavior cannot be lost, stolen, guessed, or brute-forced. Furthermore, using behavior to authenticate reduces the cognitive burden placed on users, thereby improving the usability of the authentication procedure. Although behavior has been explored as an authentication mechanism by previous researchers [21, 4, 3], this work is the first to explore the benefits of using gameplay to collect discriminative behavior on mobile touchscreen devices via a substantial user study.

To evaluate our approach we conducted a study with 30 participants who interacted with three pre-selected games and an application which did not involve any gameplay elements. This data was then processed to extract features which were useful in differentiating between users. We applied a multiclass Support Vector Machine (SVM) learning algorithm as well as a one-class SVM variant with different kernels and parameters to assess the discriminative power of the selected games. Our proposed system is capable of performing authentication in fewer than 5 seconds of gameplay with at most one false positive per day with 95% confidence and is not influenced by a user's skill or experience playing a particular game. These results suggest that using gameplay as a behavioral biometric is an effective means of authenticating users to their mobile devices. However, as not all games performed equal well in terms of authentication accuracy, care must be taken to select a game which is beneficial to the authentication process by encouraging users to make frequent distinctive gestures.

The remainder of the paper is organized as follows: Section 2 summarizes the related work. Section 3 discusses our threat model, study design, feature selection and data analysis. Section 4 presents the outcomes of data modeling and survey analysis, and Section 5 concludes the paper and presents potential future work.

2 Related Work

The most popular authentication methods are often the most straightforward to perform. This explains the widespread use of passwords, graphical patterns, and fingerprint recognition as authenticators. Since these mechanisms are the most broadly deployed, they are also some of the most studied and attacked. Weak passwords are vulnerable to guessing and dictionary attacks. To provide sufficient security, a lengthy combination of alphanumeric and special characters are required [22], which are difficult and tedious to enter on

small touch devices [16]. Fingerprint recognition has recently gained popularity as scanning hardware has been included on more smartphones and the process offers fast user identification. However, it remains vulnerable to fingerprint spoofing attacks [10]. Graphical patterns are convenient for users but susceptible to shoulder surfing and other observation attacks [16].

Behavioral biometrics, which function by analyzing patterns of user activity, have recently been gaining traction in studies as an alternative authentication method for mobile devices. Previously proposed applications include continuous behavioral authentication on mobile devices via touchscreen usage [21] and application habits [15]. These methods apply machine learning to user interactions with the mobile device to generate a model which is then used authenticate users. For example, in [4], Frank et. al used k-nearest-neighbor clustering and SVMs to classify users while they performed reading and image-viewing tasks on a mobile device. Though the time-to-detection of their scheme is unclear, their results indicated that touchscreen biometrics were suitable as one component of a broader multi-modal authentication scheme.

Khan and Hengartner empirically evaluated the device-centric nature of implicit authentication schemes in [6] and concluded that application-centric implicit authentication schemes provide significant security improvements compared to their device-centric counterpart. However, this delegation increases the development overhead of the application provider. In [13], Neal et. al surveyed over a hundred biometric approaches to mobile security and found that physiological and behavioral modalities reduce the need for remembering passwords and PINs. They concluded that these methods offered improved security for mobile devices, even though biometric security remains a complex procedure due to hardware limitations, inconsistent data, and adversarial attacks.

Feng et. al incorporated contextual application information to improve user authentication for mobile devices in [3]. With extensive evaluation, they found that their context-aware implicit authentication system achieved over 90% accuracy in real-life naturalistic conditions with only a small amount of computational overhead and battery usage. Krombholz, Hupperich, and Holz evaluated a pressure sensitive authentication method for mobile devices in [9]. Their work demonstrated that using touch pressure as an additional dimension lets users select higher entropy PIN codes that are resilient to shoulder surfing attacks with minimal impact on usability and error rates. This is contrary to what Khan et. al presented in their study, however, as hidden features like finger pressure, angular velocity, and the finger width making contact with the screen are hard to imitate via shoulder surfing.

Khan, Hengartner, and Vogel presented the results of their two-part study on usability and security perceptions of behavioral biometrics in [7] and found that 91% of participants felt that implicit authentication was convenient (26% more than explicit schemes). 81% perceived the provided level of protection satisfactory with only 11% concerned about mimicry attacks. On the other hand, false rejects were a source of annoyance for 35% and false accepts were the primary security concern for 27%. The authors concluded that implicit authentication is indeed a meaningful approach for user authentication on mobile devices with a reasonable trade-off in terms of usability and security.

In [1], Buscheck et. al discuss opportunities for improving implicit authentication accuracy and usability by including spatial touch features and using a probabilistic framework in their authentication scheme to handle unknown hand postures, showing a 26.4% reduction in the classification Equal Error Rate (EER) to 36.8%. Harbach et. al investigated

users' mobile device locking and unlocking behavior in [5] and found that on average, participants spent around 2.9% of their smartphone interaction time authenticating their device. Participants that used secure lock screens considered it unnecessary in 24.1% of situations. In their study, shoulder surfing was perceived to be a risk in only 11 of 3410 sampled situations.

Khan et. al also studied shoulder surfing and offline training in [8], which they consider to be targeted mimicry attacks. The authors evaluate the security of implicit authentication schemes and demonstrate that it is surprisingly easy to bypass them, but only if the attacker is a malicious insider who is able to observe their victims' behaviors or if the device is compromised to collect and transmit a user's touch events which can then be used to train and mimic the victim's behavior. In [2], Cherapau et. al presented their investigation of the impact Apple's "TouchID" had on passcodes for unlocking iPhones. Their study revealed no correlation between the use of TouchID and the strength of users' passcodes. The researchers also found that the average entropy of passcodes was 15 bits, corresponding to only 44 minutes of work for an attacker to find the correct password by brute force.

A shortcoming of previously proposed biometric solutions is that they often require a long time window for model construction and user authentication. To address this issue, we propose the utilization of gameplay characteristics as an authentication method. The correct choice of game can be used to encourage users to perform more distinctive gestures at a faster rate, reducing training time as well as the overall time taken to complete the authentication process. Furthermore, unlike many traditional biometric authentication methods which require specific hardware to operate, our approach is applicable to any device with a touch-sensitive screen and is thus deployable to a broad array of mobile devices. The entertainment value provided by games encourages user engagement, which is useful for the training portion of the modeling process. An added benefit of using gameplay as an authenticator is that games do not typically involve revealing any potentially sensitive user information.

This work is a continuation of a previous pilot study intended to explore the potential of games as an authentication method [17] which found the approach to be promising, but lacked sufficient data to draw statistically relevant conclusions. In this paper, we expand the scale of the study to 30 participants and perform a more rigorous assessment of its viability. The reported accuracy of the multi-class models from the pilot study are higher than those reported in this research because the models were trained on a much smaller dataset for each game, resulting in overfitting. Moreover, in order to compare the classification performance of our gameplay models against activities that do not involve gameplay, we introduced a screen without any game-based prompting to our study process.

3 Evaluation

3.1 Threat Model

In this paper, we concentrate on the user-to-device authentication process which is used to protect the sensitive data stored on a mobile device in the event that it is acquired by an unauthorized individual, such as when a device is intentionally stolen, acquired by a co-worker, or forgotten in a public place. Our solution is to require potential users to play a short, specific touchscreen game on the "lock screen" to gain access to the device.

The mechanism is intended to discriminate between an authentic user and an adversary, assuming that there is no vulnerability in the OS which may be exploited by the attacker to bypass the authentication procedure. Our threat model does not consider cases where an attacker has the time, access, and skill necessary to disassemble a device in order to manipulate its memory or directly access the data on its disk. Remote attacks via exploits and social engineering are also outside the scope of our proposed solution's threat considerations. Lastly, our model also assumes that an attacker is not able to observe or track a user's gameplay interactions and then effectively recreate them in order to impersonate the user's gestures. The issue of mimicry attacks will be addressed in future research.

3.2 Sensor Design

For our study, we developed a TouchScreen Monitor application to log users' touch interactions on Android devices. Because the Android Application Sandbox isolates data between different applications, touchscreen interactions with a particular application are not permitted to be recorded by other applications. To overcome this limitation, our proof-of-concept TouchScreen Monitor gathers raw touch screen data from the Android system using rooted access; in a practical deployment, the game used to authenticate users would be included as a built-in system lock screen. The sensor application has been developed using Java and the Android SDK framework.

The TouchScreen Monitor first executes the system command "su" to acquire root privileges. After permission is successfully granted, the application executes the system command "getevent" to record all touch events with the screen. Since raw touches are recorded at a fast sampling rate - less than 0.1 ms apart - during the logging process, they are buffered and written in bulk to the device's disk in order to minimize the number of writes performed. To improve reliability, the touchscreen log feature is implemented as a separate Android service on a different thread so it is automatically restarted to continue logging events if an error occurs in the main application. Using a separate thread for logging avoids interference with the application which is being monitored. The TouchScreen Monitor also supports uploading the collected data to a server for further analysis.

3.3 Experimental Study Design

We selected three popular games from the Google Play Store to conduct our experiments with: Angry Birds, Flow Free and Fruit Ninja. These games were chosen because they are popular unpaid games and demonstrated promising results in our pilot study [17]. Each of the three selected games also has relatively simple gameplay and gentle learning curves, which make them suitable for a diverse set of users. For the non-gameplay portion of our study, we asked users to make arbitrary gestures on a blank screen, allowing them make any type and number of interactions without any gameplay prompting. All experiments were conducted on the same Android device, which was a Samsung Galaxy S3 smartphone; exploring the applicability of games to establishing cross-device biometric profiles is another intended area of future research [14].

We designed our experiment as a within-subjects study in which volunteers were asked to play the three aforementioned games and use the blank screen sequentially, performing each activity for 5 minutes. Prior to each segment of a study session, if a participant

had never played a particular game before, they would be allowed to play the game for a few minutes in order to acclimate themselves to the gameplay requirements and controls. During the experiment, the TouchScreen Monitor was run silently in the background to record all the user interactions which occurred during each activity. For Angry Birds and Flow Free, which require the user to play the game level by level, users were required to start from the first level of each game. Since each level has different scenarios and difficulty levels, this requirement ensured that differences between users' gestures are not caused by variations in level design, but are rather introduced naturally by users in response to the same game prompting. Fruit Ninja does not follow this pattern because users play the game until they fail and start again. In this game, a score is used to assess how well users play rather than level progression.

For the non-gameplay "blank screen" task, users could interact with the blank screen in any way they wanted, with nothing displayed in order to influence them towards making particular gestures. Users were asked to perform the study tasks naturally without pressure or monitoring from the study administrator. During each session, the administrator was careful not to mention the security implications of the study in order to avoid potential priming effects. After a participant performed each of the four activities, they were asked to complete a post-conditional questionnaire. This survey contained questions which collected basic demographic information as well as information pertaining to users' experience with smartphones, video games, and mobile games in general, as well as their prior experience with each of the mobile games used in the study.

The study was advertised at our institution via fliers and in-class announcements. We recruited 30 participants in total. Because our study was conducted at a university, our survey revealed a younger participant age than is representative of the broader population, with most of the participants being students between the ages of 18 and 34. A study with a more representative pool of volunteers is a target of future research.

3.4 Feature Extraction

The raw logs collected by the TouchScreen Monitor represent atomic, low-level user interactions with the touchscreen. We extracted higher-level features from those logs to create potentially distinctive characteristics for classification. There are two approaches to extract high-level features from touchscreen usage data: parse the continuous gesture into individual points, or combine them to form aggregate swipe gestures. As we experimented, the first approach gave inferior classification performance as it does not capture some of the important characteristics of a high-level swipe, such as the speed and initial and final coordinates of the gesture.

We followed the second approach and extracted seventeen high-level features of each swipe gesture which had been demonstrated to be conducive to user classification by previous work [4, 17], including: (1) the initial X coordinate of the gesture, (2) the initial Y coordinate of the gesture, (3) the final X coordinate of the gesture, (4) the final Y coordinate of the gesture, (5) the time period during the gesture, (6) the average area covered by finger during the gesture, (7) the average finger width contacting the screen during the gesture, (8) the length of the gesture along the X axis, (9) the length of the gesture along the Y axis, (10) the distance traveled during the gesture, (11) the direction of the gesture, (12) the speed along X axis of the gesture, (13) the speed along the Y axis of the gesture, (14) the speed along the gesture's trajectory, (15) the velocity of the

gesture, (16) the angular velocity of the gesture, (17) the finger orientation change during the gesture.

3.5 Feature Analysis

In practice, some features have more discriminative power than others features. In order to measure how well these features can discriminate between users, we utilize a measurement known as the Fisher function [19]. The scalar Fisher score for each feature is defined as the ratio between the between-class variance and the sum of all within-class variances:

$$f = \frac{\sigma_b^2}{\sum_{i=1}^{i=n} \sigma_i^2}$$

where σ_i^2 is the within-class variance. σ_b^2 is the between-class variance, which is defined as:

$$\sigma_b^2 = 1/n \sum_{i=1}^{i=n} (\mu_i - \mu_g)^2$$

where μ_i is the statistical mean for the feature values of user i, and μ_g is the grand mean of all mean values μ_i.

A higher between-class variance indicates that a feature is more distinctive for each user. A lower within-class variance implies that a feature's values are more consistent for the same user. Thus, features with lower relative Fisher scores can be considered potentially redundant and candidates for removal to optimize classification performance.

3.6 Data Modeling and Analysis

We implemented R language scripts to apply a multiclass Support Vector Machine (SVM) to the extracted feature set to classify participants using a variety of kernels and parameters. We choose to explore a SVM for gameplay authentication because it is a well-understood algorithm which had been successfully applied to behavioral authentication in the past, which allowed our experiments to focus on the question of the applicability of gameplay to the task of authenticating users. For the SVM implementation, we utilized the LibSVM based "e1071" R package. We conduct multiclass SVM classification with C-Support Vector Classification (C-SVC) using Radial Bias Function (RBF) and Polynomial kernel functions. To achieve multiclass classification, the classifier applied a "one-versus-one" technique in which binary classification is applied to each pair of users. 10-fold cross validation is used to conserve data while training and testing our model.

Some of the performance gains associated with a particular task could potentially be caused by the availability of larger quantities of training data. To prevent this factor from influencing our results, we do not train the model on each user's full dataset due to the fact that some activities cause users to make many more gestures than other activities and not all users performed precisely the same number of gestures during each task. For example, the Flow Free task resulted in nearly three times as many gesture samples to work with relative to Angry Birds. In our experiment, all users play the same initial levels for each game. However, since Flow Free does not consume as much time with animations as Angry Birds, users are able to play it at a faster pace and make more gestures. For an unbiased

comparison, we construct the training dataset by choosing an equal number of samples per each user across all activities, in which the smallest amount of gestures generated by any user across all activities is the sample size for each user. For better classification performance, the data is standardized to have a mean of zero and a standard deviation of 1 before modeling.

To measure and compare the performance of each task and modeling technique, we plotted Receiver Operator Characteristic (ROC) curves and calculated the Area Under the ROC Curve (AUC). An ROC curve is a plot of a false positive rate (FPR) on the X axis against the classifier's true positive rate (TPR) on the Y axis which is generated by varying the acceptance threshold used in the classification process. The intersection point of the curve with the Y-Axis has a FPR of 0%, which causes classification to be highly restrictive in terms of FPR and does not allow any misclassification of illicit users as authorized users. Similarly, the point of the ROC curve which has a true positive rate of 1 will accept all authentic users but may decrease the rate of rejecting an unauthorized user. An ideal classifier is a model in which the FPR is 0% and the TPR is 100%, resulting in an AUC of 1. However, this ideal model is often impossible to achieve. Therefore, in practice, there is always a trade-off between the classifier's TPR and FPR; that is, a threshold that causes a classifier to have a lower FPR also has a lower TPR. On the other hand, a threshold that increases the TPR will also decrease the FPR. Our goal is to maximize the AUC, which represents the maximization of the chance of successfully authenticating a legitimate user while minimizing the rate of accepting unauthorized users.

Similarly, a Detection Error Tradeoff (DET) curve plots a classifier's FPR against its false negative rate (FNR), which is used to visualize the relationship between these errors. A classifier's FNR is related to its TPR via the equation: $FNR = 1 - TPR$. The Equal Error Rate (EER), which is the common value at which the FNR and FPR are equal, is used to express the balance between the false acceptance and false rejection performance of a classifier, with a lower EER corresponding to more accuracy in the classifier.

3.7 One-Class Classification

In addition to multiclass SVM classification, we also implemented a one-class SVM (oc-SVM) in R. With a multiclass classification approach, each user model is trained using both positive examples of their own data as well as negative examples from other users' data. In contrast, a one-class method only trains models using positive examples of each user's authentic data. A one-class approach is more appropriate to the task of user authentication, where the goal is to discern whether the legitimate device owner is using the device or any other user is, rather than determining which specific user is controlling it. Another practical reason why one-class modeling is more suitable for authentication is that a particular device would not have direct access to another user's model, and even if this information could be shared, the process would be difficult to scale.

For this purpose, we implemented a oc-SVM in R using the LibSVM library with a "one-classification" kernel type which accepts only positive data of an authentic user when training. The oc-SVM is applied for all three games as well as the "blank screen" task, just as with the multiclass modeling process. For each activity, we applied an oc-SVM to create a separate model for each user, in which 80% of their data is used to train the model. To validate the classifier, the remaining 20% of a user's data is combined with equal samples of every other users' data to create the validation set. As was the case with multiclass

SVM classification, AUC and EER are again used as metrics to assess the accuracy of the oc-SVM models.

3.8 Survey Analysis

Users were asked to complete a survey at the conclusion of each experimental session. The post-conditional survey posed questions regarding demographic information, mobile device experience, video game experience, mobile game experience, and how engaged with each game people felt. The specific queries compromising the questionnaire are presented in Appendix A. Responses to these questions allowed us to categorize users according to different attributes in order to infer information about what aspects of participants' backgrounds may have an effect on gameplay based authentication accuracy. For example, users were asked how much experience they had which each of the three games used in our study: one week or less, one month, three months, six months, one year, or more than a year. This survey item provided insight into whether user classification, and thus authentication, was more or less accurate for users with a lot of experience playing a particular game as opposed to users who had not played the game very much, if at all. To answer this question, we grouped the oc-SVM AUC results according to gameplay experience and applied a one-way Analysis of Variance (ANOVA) test to assess how statistically significant differences in classification performance were between each gameplay experience group.

4 Results

4.1 Multiclass Classification

As detailed in Section 3.6, we implemented R scripts using the LibSVM library to perform multiclass SVM classification with the C-Support Vector Classification (C-SVC) training algorithm and tested both RBF and polynomial kernel functions. For the polynomial kernel, we conducted tests using different combinations of polynomial degrees and C parameter values, which control the size of the hyperplane margin. Based on our experiments, a C parameter of 10 resulted in the most accurate model. To optimize the performance using the RBF kernel, we applied hyperparameter optimization by varying the value of gamma from 0.1 to 0.9 and performed model training and testing for each gamma value. A gamma value of 0.51 produced the lowest error rates for this type of kernel.

After settling on modeling parameters, we plotted ROC curves which captured the classification performance for each user and activity. First, the classification probability that a validation instance belongs to a user is calculated. Next, ROC curves are generated by varying the acceptance threshold applied to these probability values. These ROC curves were used to calculate the AUC for each user and activity. The individual per-user AUC values were averaged across all users to produce an aggregate AUC value for each task. We followed a similar process to derive average EER values for each study task. First, DET curves were plotted for each user and task. These DET curves were used to find the EER value for each user and task combination, and these per-user EER values were then averaged together to produce one overall EER per task.

We calculated the AUC and EER values to facilitate a comparison between how well our classifier was capable of distinguishing between users based on the touchscreen gestures

they made while playing each game as well as the unprompted "blank screen" task. Table 1 presents the average AUC and EER of the three games - Angry Birds, Flow Free, and Fruit Ninja - as well as the "blank screen" activity. As shown in Table 3, the average AUC using a SVM with a RBF kernel is over 0.9 for Angry Birds and Fruit Ninja. The Angry Birds and Fruit Ninja gameplay resulted in better classification performance than the "blank screen" task, which had no gameplay context. Our hypothesis is that this was caused in part because users found making gestures without prompting from a game to be tedious, as our post-conditional survey responses revealed that only 12.5% of participants felt engaged during this activity.

The activity which resulted in the highest classification error rates was Flow Free. We conclude that the most logical explanation for this result was due to the nature of Free Flow's gameplay, in which each level is a puzzle which typically has one specific solution. Thus, all users are required to make very similar gestures to complete each level, which made it more difficult to differentiate between each user's gameplay habits. The level of engagement among study participants may have also played a role in the relatively low modeling accuracy observed for Flow Free gameplay. According to our survey feedback, 59.4% of study participants felt engaged while playing Flow Free, whereas the percentage is 65.6% for Angry Birds. The most engaging game was Fruit Ninja with 78.10% of participants responding that they were engaged by its gameplay. When taken as a whole, these results imply that designs of Angry Birds and Fruit Ninja do the most to encourage users to make distinctive gestures. We also note that the accuracy of SVM classification is consistently higher when using a RBF kernel in comparison to a polynomial kernel for all four study tasks.

Activity	**SVM Kernel**	**Average AUC**	**Average EER**
Angry Birds	Polynomial	0.870	20.38%
Angry Birds	RBF	**0.963**	**10.34%**
Flow Free	Polynomial	0.734	32.79%
Flow Free	RBF	**0.804**	**27.31%**%
Fruit Ninja	Polynomial	0.869	21.03 %
Fruit Ninja	RBF	**0.919**	**15.64%**
Blank Screen	Polynomial	0.847	23.53%
Blank Screen	RBF	**0.898**	**18.33%**

Table 1: Multiclass SVM Classification Results for All Activities

4.2 Feature Analysis

Table 2 contains a list of the 17 features we considered during our study and their corresponding Fisher scores. These values were calculated based on the gesture feature vectors extracted from all users across all activities. The features are arranged by Fisher score in descending order. Table 2 also lists another feature performance metric which we refer to as the "classification contribution," which is meant to capture the impact of omitting the feature on modeling performance. To determine the classification contribution of each feature, we implemented an R script to iterate over the feature set, remove each feature

Feature	Fisher Score	Classification Contribution
Average Finger Width	0.003095	0.26%
Time Period	0.002860	1.16%
Average Area Covered	0.002793	0.67%
Initial X Coordinate	0.001341	0.51%
initial Y coordinate	0.001184	0.63%
Angular Velocity	0.000884	0.66%
Length along X Axis	0.000764	-0.01%
Length along Y Axis	0.000720	-0.11%
Distance Traveled	0.000705	0.48%
Speed along Y Axis	0.000416	-0.03%
Velocity	0.000416	-0.04%
Final Y Coordinate	0.000351	0.21%
Speed along X axis	0.000343	0.05%
Finger Orientation Change	0.000302	0.74%
Final X Coordinate	0.000271	0.17%
Trajectory Speed	0.000078	-0.03%
Direction	0.000059	-0.12%

Table 2: Fisher Scores for Features Across All Gameplay Activities

one at a time, apply multiclass SVM classification with the given feature removed, and calculate the AUC value produced when each feature is left out. The classification contribution value for each feature is obtained by subtracting the AUC after removing the feature from the AUC which is achieved when modeling is performed using all available features.

To verify the accuracy of our Fisher scores, we calculated the Pearson correlation coefficient between the Fisher scores and the classification contributions. The Pearson correlation coefficient measures the linear dependence between two variables, where a result of 1 indicates a complete positive linear correlation, while a value of -1 implies a totally inverse linear correlation, and a 0 implies no correlation between variables. The Pearson correlation coefficient between our features' Fisher scores and classification contributions is 0.6, which suggests that the classification contributions are highly correlated with Fisher scores. Thus, features with a lower Fisher score also tend to make less of a contribution to classification performance, which supports the accuracy of our estimates of each feature's discriminative power. In our dataset, features pertaining to gesture direction and speed tend to have the lowest Fisher score and classification contribution, which identifies these features as potentially redundant and therefore good candidates for removal in order to streamline our model.

4.3 One-Class Classification

After completing our multiclass modeling experiments, we repeated the classification process using a one-class modeling approach as described in Section 3.7. We again experimented with both RBF and polynomial kernel functions during our tests. Because one-class modeling is more appropriate to the application of mobile authentication, performance curves have been included in addition to a result summary. Figures 1, 3 and 5

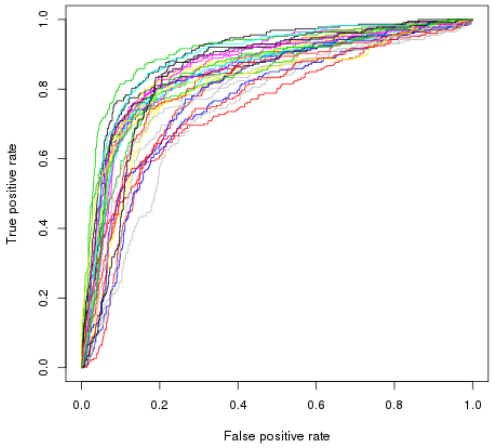

Figure 1: Per-User ROC Curves for oc-SVM Model of Angry Birds Gameplay

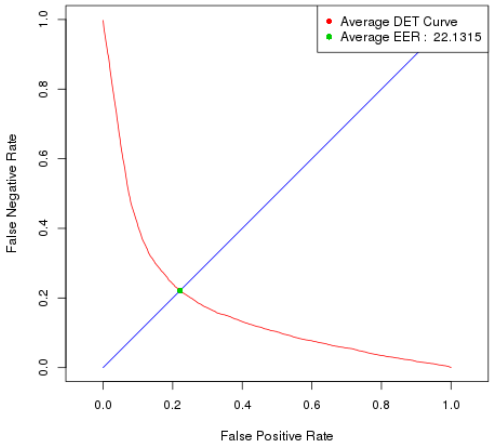

Figure 2: Average DET Curve for oc-SVM Model of Angry Birds Gameplay

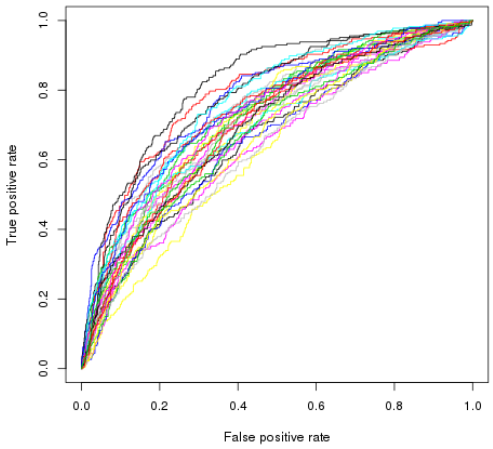

Figure 3: Per-User ROC Curves for oc-SVM Model of Flow Free Gameplay

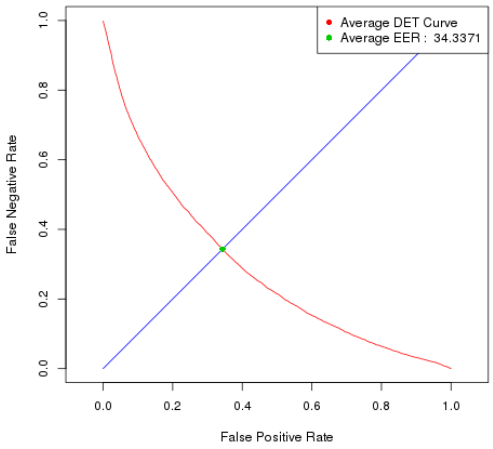

Figure 4: Average DET Curve for oc-SVM Model of Flow Free Gameplay

present ROC curves for Angry Birds, Flow Free and Fruit Ninja gameplay classification. Figures 2, 4 and 6 present the DET curves which resulted from classifying users' touch-screen activity with these games. These figures were the result of using an RBF kernel during one-class modeling, which again produced models with lower error rates relative to the polynomial kernel function. Table 3 summarizes the performance of these models by presenting the average AUC and EER across all users for each study task and kernel type. We experimentally determined the model parameters which minimized the error rates of our classifier and found that a gamma value of 0.802, which controls the variance of the

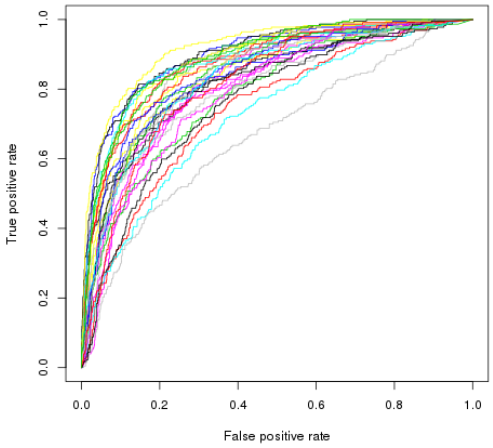

Figure 5: Per-User ROC Curves for oc-SVM Model of Fruit Ninja Gameplay

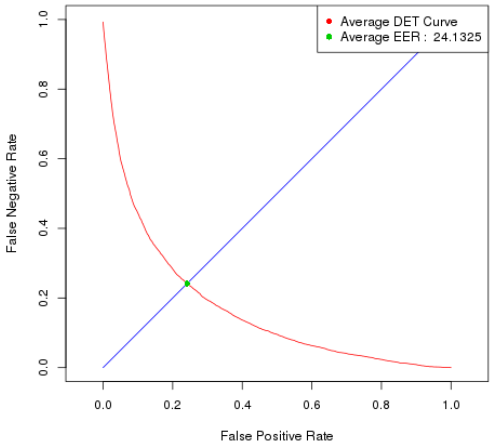

Figure 6: Average DET Curve for oc-SVM Model of Fruit Ninja Gameplay

kernel, and a nu value of 0.1608, which controls the amount of permissible training errors, resulted in the best performance.

The oc-SVM classification results largely mirror those produced by multiclass classification. We observe that for all tasks and kernels, the oc-SVM produced less accurate classification results than the multiclass classifier. This is best explained by the removal of negative samples during the training process, which makes it more difficult for the model to learn the "boundary" between positive and negative instances. For both multiclass and one-class modeling, Angry Birds and Fruit Ninja gameplay resulted in more accurate user classification relative to the unprompted "blank screen" task. However, Flow Free resulted in higher error rates than either other game or the unprompted gestures.

These results suggest that touchscreen patterns while playing computer games can be used to differentiate between users, and thus can be applied as an authentication mechanism for mobile devices. However, this result is not generalizable to all games. The style of gameplay must be considered when selecting a game to utilize as an authentication task. We hypothesize that Fruit Ninja and Angry Birds resulted in better classification performance because they prompted users to quickly make touchscreen gestures which were highly consistent for each user while being very distinctive between users. Both games encourage users to interact with the touchscreen in a very free-form fashion. In contrast, Flow Free demonstrated worse classification performance relative to the unprompted "blank screen" task because it forces users to make very specific touchscreen gestures in order to progress through the game. This resulted in study participants making very similar gestures to one another at a slower rate, which made it more challenging to distinguish users from each other. To summarize, gameplay characteristics must be taken into account when designing a behavioral authentication system which leverages a computer game to improve classification performance.

Activity	SVM Kernel	Average AUC	Average EER
Angry Birds	Polynomial	0.521	49.85%
Angry Birds	RBF	**0.832**	**22.13%**
Flow Free	Polynomial	0.507	49.28%
Flow Free	RBF	**0.712**	**34.34%**
Fruit Ninja	Polynomial	0.493	51.39 %
Fruit Ninja	RBF	**0.831**	**24.13%**
Blank Screen	Polynomial	0.477	53.60%
Blank Screen	RBF	**0.806**	**39.12%**

Table 3: oc-SVM Classification Results for All Activities

4.4 Effect of Experience

If a game is used to authenticate users, a natural question is the extent to which experience playing the game impacts its effectiveness. The authentication game should not require a particular amount of skill and should be capable of classifying novices just as well as experts. Conversely, getting better at playing a particular game should also not result in a degradation of classification performance. This could be possible if the gestures of experienced players converge to a optimal "solution" for a game. We attempt to explore this question using participant's responses to our post-conditional questionnaire. One of the questions posed was how much experience a user had with each of the three games that were tested. We divided our participants into groups based on how much experience they reported playing each game and found the average AUC from applying the oc-SVM model to each experience group. These results are presented in Table 4, which shows that classification accuracy does not vary much between experience groups.

We applied a one-way ANOVA test to the AUC values for these groups in order to determine if any statistically significant differences existed between users who have spent different amounts of time playing each game. Table 5 summarizes the results of this test. F represents the ratio of the variance between and within each gameplay experience group. F-critical is the threshold for determining if a significant different exists between the data groups under consideration, which we calculated using a 95% significance level. Since the F value is well below the F-critical value for each game, the null hypothesis of the ANOVA test can be accepted, which indicates that no statistically significant differences exist between the classifier's performance on each experience group. We therefore conclude that the amount of experience a user has playing a particular game does not effect the accuracy of using the game to authenticate them. Game-based authentication is thus equally applicable to users of all levels of experience with a game.

4.5 Time Taken to Authenticate

Since unlocking a mobile device is such a frequent activity, an important aspect of the usability of a mobile authentication scheme is the amount of time it takes to complete. To determine how long it takes to use a game to authenticate a user on a mobile device, we must first determine an acceptable threshold of false positives; that is, how frequently is the authentication scheme allowed to incorrectly identify a legitimate user as an attacker? We settled on one false positive per day as a reasonable threshold. According to the results

Experience	Angry Birds	Flow Free	Fruit Ninja
Never	0.843	0.723	0.823
One week or less	0.819	0.680	0.835
1 month	-	0.728	0.841
3 month	0.846	0.696	0.859
6 month	-	0.724	-
1 year	0.844	0.730	0.844
Over 1 year	0.848	0.660	0.841

Table 4: Average oc-SVM AUC Categorized by Amount of Experience with Each Game

Activity	F	F-critical
Angry Birds	0.623	2.759
Flow Free	0.929	2.528
Fruit Ninja	0.187	2.621

Table 5: ANOVA Results for Each Game

of a recent study, an average smartphone user unlocks his or her mobile device an average of 110 times per day [20]. Allowing for one false positive per day would thus translate into a false positive rate of $1/110 = 0.91\%$ per each one second sample of touchscreen gameplay behavior. At this low false positive rate, using an oc-SVM based on Angry Birds as an authentication game would result in a true positive detection rate of 49.88% per sample. Thus, for each second of gameplay there is a 50.12% of failing to detect that the game is being played by someone other than the legitimate device owner. Detecting device misuse with 95% confidence would thus require 5 gameplay samples:

$$\begin{aligned} 0.5012^x &< (1 - 0.95) \\ 0.5012^x &< 0.05 \\ x &> 4.34 \end{aligned}$$

Thus, 5 seconds of Angry Birds gameplay activity can be used to authenticate users with 95% accuracy and at most one false positive per day. Though slower than authentication via traditional biometrics such as fingerprints, a 5 second time interval is reasonable in the context of mobile authentication. This suggests that gameplay can be utilized to reduce the time required to authenticate users via biometrics based on touchscreen behavior.

5 Conclusion

To summarize, this paper presented a novel approach to mobile authentication in which users are asked to play a game in order to authenticate themselves to their mobile devices. Computer games are potentially beneficial to the authentication process as they are usable by design and encourage players to rapidly make unique touchscreen gestures. To assess the viability of this proposed approach, a study was conducted in which 30 users were asked to play three popular mobile games as well as perform touchscreen gestures without

gameplay prompting. Features which captured users' gameplay habits were extracted from these gestures and modeled using SVMs. Our results indicate that games are potentially useful authenticators. A multiclass model based on the Angry Birds game resulted in an AUC of over 0.95 and an EER of 10.34%. A more practical one-class model of Angry Birds gameplay was shown to be capable of detecting device misuse in 5 seconds with 95% accuracy and one false positive per day. We conclude that authenticating users based on the manner in which they play a game can improve the performance of authentication relative to touchscreen tasks which do not involve gameplay. However, the game used as an authentication mechanism must be selected with care. Games which encourage users to make a wide variety of distinctive gestures were found to be beneficial, while those which required slow and specific gestures were not. Experience playing a game was found to not have an impact on the accuracy of authentication.

This work demonstrates the plausibility of using computer games for mobile authentication. However, future exploration is required to answer a number of remaining questions regarding gameplay-based behavioral biometric authentication. As future work, we intend to perform studies with larger, more representative volunteer groups in order to explore the susceptibility of gameplay authentication to mimicry attacks in which an adversary attempts to replicate a legitimate user's gameplay habits. We also plan to assess the extent to which gameplay behavior is affected by device hardware and firmware. We will further consider which characteristics of gameplay are conducive to user classification in order to more fully examine the usability of game-based authentication and the extent to which it can be generalized.

Acknowledgements

Many thanks to Graduate Assistant Tuan Ngyuen for his efforts performing the study reported in this paper and Graduate Assistant Sheharyar Naseer for his editing assistance.

References

1. D. Buschek, A. De Luca, and F. Alt. Improving Accuracy, Applicability and Usability of Keystroke Biometrics on Mobile Touchscreen Devices. In *Conference on Human Factors in Computing Systems (CHI)*, pages 1393–1402, 2015.
2. I. Cherapau, I. Muslukhov, N. Asanka, and K. Beznosov. On the Impact of Touch ID on iPhone Passcodes. In *Symposium on Usable Privacy and Security (SOUPS)*, pages 257–276, 2015.
3. T. Feng, J. Yang, Z. Yan, E. M. Tapia, and W. Shi. TIPS: Context-Aware Implicit User Identification Using Touch Screen in Uncontrolled Environments. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications (HotMobile)*, page 9, 2014.
4. M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. *Transactions on Information Forensics and Security (TIFS)*, 8(1):136–148, 2013.
5. M. Harbach, E. Von Zezschwitz, A. Fichtner, A. De Luca, and M. Smith. Its a Hard Lock Life: A Field Study of Smartphone (Un)Locking Behavior and Risk Perception. In *Symposium on usable privacy and security (SOUPS)*, pages 9–11, 2014.
6. H. Khan and U. Hengartner. Towards Application-Centric Implicit Authentication on Smartphones. In *Workshop on Mobile Computing Systems and Applications (HotMobile)*, page 10, 2014.

7. H. Khan, U. Hengartner, and D. Vogel. Usability and Security Perceptions of Implicit Authentication: Convenient, Secure, Sometimes Annoying. In *Symposium on Usable Privacy and Security (SOUPS)*, pages 225–239, 2015.
8. H. Khan, U. Hengartner, and D. Vogel. Targeted Mimicry Attacks on Touch Input Based Implicit Authentication Schemes. In *International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 387–398, 2016.
9. K. Krombholz, T. Hupperich, and T. Holz. Use the Force: Evaluating Force-Sensitive Authentication for Mobile Devices. In *Symposium on Usable Privacy and Security (SOUPS)*, pages 207–219, 2016.
10. S. R. Labs. Fingerprints are Not Fit for Secure Device Unlocking. Available at: `https://srlabs.de/bites/spoofing-fingerprints/`, 2014. Retrieved 12/18/17.
11. I. Lookout. Phone Theft In American, Breaking down the phone theft epidemic. Available at: `https://transition.fcc.gov/cgb/events/Lookout-phone-theft-in-america.pdf`, 2014. Retrieved 12/18/17.
12. A. Murdock. Consumers Spend More than 1 Billion Hours a Month Playing Mobile Games. Available at: `http://www.vertoanalytics.com/consumers-spend-1-billion-hours-month-playing-mobile-games`, 2015. Retrieved 12/18/17.
13. T. J. Neal and D. L. Woodard. Surveying Biometric Authentication for Mobile Device Security. *Journal of Pattern Recognition Research*, 1:74–110, 2016.
14. T. Ngyuen and J. Voris. Touchscreen Biometrics Across Multiple Devices. In *Who Are You?! Adventures in Authentication Workshop (WAY) co-located with the Symposium on Usable Privacy and Security (SOUPS)*, 2017.
15. M. B. Salem, J. Voris, and S. Stolfo. Decoy Applications for Continuous Authentication on Mobile Devices. In *Who Are You?! Adventures in Authentication Workshop (WAY) co-located with the Symposium on Usable Privacy and Security (SOUPS)*, 2014.
16. F. Schaub, R. Deyhle, and M. Weber. Password Entry Usability and Shoulder Surfing Susceptibility on Different Smartphone Platforms. In *Conference on Mobile and Ubiquitous Multimedia (MUM)*, 2012.
17. P. Scindia and J. Voris. Exploring Games for Improved Touchscreen Authentication on Mobile Devices. In *Who Are You?! Adventures in Authentication Workshop (WAY) co-located with the Symposium on Usable Privacy and Security (SOUPS)*, 2016.
18. D. Tapellini. Smart Phone Thefts Rose to 3.1 Million in 2013. Available at: `http://www.consumerreports.org/cro/news/2014/04/smart-phone-thefts-rose-to-3-1-million-last-year/index.htm`, 2014. Retrieved 12/18/17.
19. M. Welling. Fisher Linear Discriminant Analysis. *Technical Report, Department of Computer Science, University of Toronto*, 2005.
20. V. Woollaston. How Often Do You Check Your Phone? The Average Person Does it 110 Times a DAY (And up to Every 6 Seconds in the Evening). Available at: `http://www.dailymail.co.uk/sciencetech/article-2449632/How-check-phone-The-average-person-does-110-times-DAY-6-seconds-evening.html`, 2013. Retrieved 12/18/17.
21. H. Xu, Y. Zhou, and M. R. Lyu. Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones. In *Symposium On Usable Privacy and Security (SOUPS)*, 2014.
22. J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password Memorability and Security: Empirical Results. *IEEE Security & Privacy*, 2004.

A Appendix: Study Questionnaire

Table 6 lists the survey questions that were used in our study in the order they were presented to participants.

Number	Question
1	What is your age?
2	What is your gender?
3	What is your ethnicity? (Please select all that apply.)
4	What is the highest level of education you have completed?
5	Have you ever used a mobile device (such as smartphones, tablets, ebook readers, or portable game systems)?
6	How many different mobile devices (such as smartphones, tablets, ebook readers, or portable game systems) have you ever used?
7	How many different mobile devices (such as smartphones, tablets, ebook readers, or portable game systems) do you currently own?
8	In a typical day, how many hours do you spend using mobile devices (such as smartphones, tablets, ebook readers, or portable game systems)?
9	What mobile operating system have you used?
10	In the past 30 days, have you used a mobile device (such as a smartphones tablet, ebook reader, or portable game system) to do any of the following activities?
11	I am an experienced mobile device user.
12	How many apps are installed on your mobile device?
13	What method do you use to unlock your mobile devices? (Please select all that apply.)
14	Have you ever played a video game?
15	In a typical day, how many hours do you spend playing video games?
16	How often do you play video games?
17	Please list some of your favorite video games.
18	Have you ever played a video game on a mobile device?
19	In a typical day, how many hours do you spend playing games on mobile devices?
20	How often do you play video games on a mobile device?
21	Please list some of your favorite games for mobile devices.
22	How long have you been playing Angry Birds?
23	How long have you been playing Flow Free?
24	How long have you been playing Fruit Ninja?
25	I felt engaged while playing Angry Birds.
26	I felt engaged while playing Flow Free.
27	I felt engaged while playing Fruit Ninja.
28	I felt engaged while interacting with the blank screen.
29	The mobile device was very responsive during the experiment.
30	The touchscreen was very responsive during the experiment.
31	I think that having to play a game before accessing my mobile device would be easier to use than my current authentication technique.
32	I think that having to play a game before accessing my mobile device would be more secure than my current authentication technique.

Table 6: Post-Conditional Study Questionnaire