# Where Usability and Security Go Hand-in-Hand: Robust Gesture-Based Authentication for Mobile Systems

**Can Liu, Gradeigh D. Clark, Janne Lindqvist**
Rutgers University

## ABSTRACT

Gestures have recently gained interest as a secure and usable authentication method for mobile devices. Gesture authentication relies on recognition, wherein raw data is collected from user input and preprocessed into a more manageable form before applying recognition algorithms. Preprocessing is done to improve recognition accuracy, but little work has been done in justifying its effects on authentication. We examined the effects of three variables: location, rotation, and scale, on authentication accuracy. We found that an authentication-optimal combination (location invariant, scale variant, and rotation variant) can reduce the error rate by 45.3% on average compared to the recognition-optimal combination (all invariant). We analyzed 13 gesture recognizers and evaluated them with three criteria: authentication accuracy, and resistance against both brute-force and imitation attacks. Our novel multi-expert method (Garda) achieved the lowest error rate (0.015) in authentication accuracy, the lowest error rate (0.040) under imitation attacks, and resisted all brute-force attacks.

### ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): Input devices and strategies; I.5.2. Pattern Recognition: Classifier design and evaluation

### Author Keywords

security; gesture; authentication; mobile device

## INTRODUCTION

The number of threats to personal information stored on mobile devices have begun to increase as these devices are more important in day-to-day life. Current authentication methods present their own set of concerns. Text-based passwords have many problems, ranging from password reuse [28, 23] to weak password selection [8]. Biometric methods have difficulties from revoking the authentication token [47] to misidentification [47] (e.g. fingerprint scanner scanning a scarred finger).

The mobile computing and systems community has recently proposed and studied several variations of gesture authentication systems [44, 13, 53, 24, 7, 25, 45, 18, 43, 55]. In the context of this work, gestures are passwords traced out on the surface of a touchscreen using one or more fingers [44] as depicted in Figure 1. Gestures have similarities to graphical password systems, such as the Android $3 \times 3$ pattern unlock [50] or the Windows 8 click-based password [54], but have not been shown to suffer from the same weaknesses. They are free from issues of the $3 \times 3$ pattern unlock: the limited password space, biased password selection, and low security (less than a 3-digit PIN) [46]. Additionally, gestures have so far not been shown to be weak against attack algorithms, in contrast to, for example, Windows 8 click-based password system [54].


**Figure 1. Example of a free-form gesture. The user in the picture has decided to use two fingers, and the corresponding gesture is drawn in different colors for visualization purposes. We note that the actual implementation does not require using visual cues in the generation or recall of the gesture.**

Any gesture-based authentication system will need to have an algorithm to interpret the users' gestures – a gesture recognizer [13]. These recognizers perform preprocessing to the data when computing results. This preprocessing strips information away from a gesture by controlling variables to make it easier to perform the recognition task. For example, a recognizer may need all gestures to be centered at the origin. Information is then lost – the original screen location – at this step. While this makes recognition easier, it is not clear what effects these preprocessing methods have on authentication. Prior work has not justified and analyzed how these preprocessing procedures can affect authentication performance.

Proposed gesture authentication systems have implemented classification methods from Support Vector Machine [7, 25, 24] to Dynamic Time Warping [40, 4, 52, 5]. These systems justify the selection of a particular method with a statement of theoretical advantages over other methods. Based on past


*CHI'17*, May 06 - 11, 2017, Denver, CO, USA

ACM 978-1-4503-4655-9/17/05...$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025879

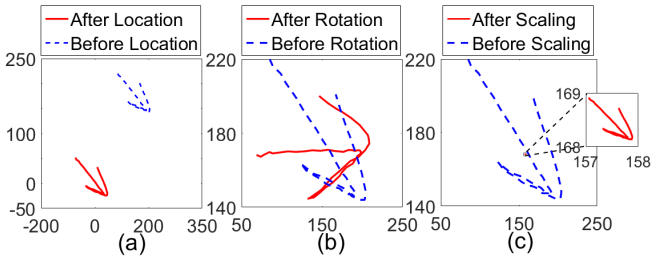


**Figure 2. Example gestures before and after separate preprocessing steps. Figure(a) shows location preprocessing; Figure(b) shows rotation preprocessing; Figure(c) shows scale preprocessing.**

work, it is hard to judge which recognition method is optimal for authentication.

Previous work on gesture-based authentication can be divided into three thrusts: continuous authentication [24] [7], template-based gesture authentication [25] [45] [18] [13] [43] [55] and user-generated free-form gestures [44, 53]. Continuous authentication is implicit; machine learning models are used to detect whether a user is using the phone a similar way each time. Template-based systems use a reduced space of pre-defined gestures that the users are asked to repeat. Continuous authentication methods are constraining – if the input device's screen size changes, the way the user interacts with the device changes. This is illustrated by the different ways people hold tablets versus smartphones. The two different modalities of interaction make it difficult to recognize the same person on one device with data from another. It is not clear what the usability and security trade-offs are. Free-form user-generated gestures can, in contrast, be used in any environment. In this work, we focus on understanding and implementing a strong novel gesture recognizer for free-form gestures.

Our major contributions are as follows:

**We are the first to present the effects of preprocessing invariances on the security and usability of gesture-based authentication systems.** We use the Protractor [32] recognizer, a popular algorithm for free-form gesture authentication [44, 53], to analyze the effects of three gesture invariances: scale, rotation, and location. Figure 2 shows examples of three distinct preprocessing steps: scale invariance (gestures are scaled down to a unit area), location invariance (gestures are translated to the same point), and rotation invariance (gestures are rotated to some constant arc). Each one can be turned on or off, leaving eight combinations in total. We examined five gesture datasets and found an authentication-optimal combination (location invariant, scale variant, and rotation variant) that reduces the error rate by 45.3% on average compared to the recognition-optimal combination (all invariant).

**We present a comprehensive study of different approaches to implementing gesture-based authentication systems.** We designed and implemented two novel Multi-Expert (ME) recognizers: **Garda** and **SVMGarda** recognizers. We also implemented eleven popular recognizer methods:

Sequences matching groups: Protractor, Edit Distance on Real sequence (EDR), Longest Common Subsequences (LCS), Dynamic Time Warping (DTW);

One-class Support Vector Machine (SVM) group: Protractor-, EDR-, LCS-, DTW- kernel based SVM;

Hidden Markov Model (HMM) group: discrete HMM (dHMM), segment HMM (sHMM), continuous HMM (cHMM).

**We evaluated the 13 recognizers on three criteria: 1) authentication performance (i.e. equal error rate) with five datasets; 2) imitation attacks with two datasets; 3) brute force attacks.** Our result shows that Garda achieved the lowest average error rate (0.015) for authentication performance, the significantly lowest average error rate (0.040) for imitation attacks, and resistance to all brute-force attacks.

## RELATED WORK

There are several approaches and analyses related to gesture authentication methods. Sherman et al. [44] introduced a novel information theoretical metric to quantify the security of free-form gestures and demonstrated a way to authenticate multi-touch gestures with Protractor [32]. They do not examine the influence of gesture variants on authentication performance. Clark and Lindqvist [13] gave a systematic rendition of how to evaluate gesture recognition methods. Yang et al. [53] studied free-form gestures and text passwords in the field and showed how gestures outperform text passwords in mobile authentication. De Luca and Lindqvist [33] gave an overview of several usability and security issues with smartphone authentication and different approaches to solve them.

Biometric features of human hands have been leveraged to perform gesture-based authentication [40]. However, this work focuses on the ability of users to perform pre-defined gestures and has high error rates (about 10%) except with user-defined gestures. Similarly, GEAT [43], used biometric features of a user's gesture like finger velocity and stroke time to distinguish different users. 3D gestures have been used for authentication as well. A system using the Microsoft Kinect, called KinWrite [45], has users perform gestures in midair to authenticate themselves. AirAuth [5] also developed a midair gesture recognition method. BoD shapes [18] are 2D gestures collected on the back of a device using two phones connected back to back. XSide [17] is a stroke-based authentication mechanism that uses front or back of smartphones.

Zheng et al. [55] designed an authentication system by recognizing user's tapping password behavior based on a list of features including acceleration, pressure, size and time collected during authentication. De Luca et al. [16] also introduced an authentication scheme based on a user's touch pattern. Burgbacher et al. [9] introduced an authentication scheme based on gesture keyboards. Shirazi et al. [41] introduced a 3D magnetic gesture recognition system. Schaub [42] examined five existing graphical password schemes and found that the design space is expressive enough to capture all aspects of a graphical password. They then give some guidelines for how to design graphical passwords.

Multi-expert systems have been applied to authentication in other context before. For example, Czyz [15] presented a multi-expert system for face authentication with sequential fusion of scores of faces' successive video frames. The final decision is combined by several face authentication schemes. Dimauro et al. [20] presented a multi-expert verification system for processing bank checks. It combines three algorithms: structure-based, component-oriented approach, and a highly-adaptive neural network based method.

Finally, the HCI community has also studied free-form and user-defined gestures from different angles. Oh et al. [36] found that user-defined gestures may be ambiguous so they implemented a mixed-initiative approach to improving gestures quality. Nacenta et al. [35] found that user-defined gestures are easier to remember than pre-designed gestures.

In summary, there has been no prior work on systematically comparing different approaches for implementing gesture recognizers for authentication. Our work contributes by implementing thirteen different approaches, compares them with multiple datasets and under different attack scenarios. Based on this analysis we have presented Garda, a novel multi-expert gesture recognition system for authentication. Our work also shows that Garda is usable on mobile devices. Finally, we present a systematic analysis of how preprocessing steps, a vital part of gesture recognition, affect the performance of gesture-based authentication systems.

## ANALYSIS METHODS

### Feature Extraction

The gestures in our datasets are collected by mobile devices. Although the sampling rates and dimensions differ, they are consistent across their datasets in that there are multiple measurements for each gesture and that the $(x, y)$ coordinates and timestamps appear in every dataset. These are the only features that were compared.

#### *Recognition Performance Metrics*

We use Equal Error Rate (EER) and the Receiver Operating Characteristic (ROC) curve to evaluate the performance of gesture recognizers for authentication. The EER is a point on the ROC curve where the False Acceptance Rate (FAR), the ratio of accepted false attempts to the total number of attempts, is equivalent to the False Reject Rate (FRR), the ratio of rejected true user attempts over the total number of attempts [22]. This represents a usability-security trade-off point: where the number of rejected true attempts equals the number of attackers permitted. The ROC curve and the corresponding FAR and FRR values reflect the behavior of an authentication method with varying thresholds.

We performed two types of attacks to evaluate the security of the recognizers: brute-force and imitation attacks. Resistance to brute-force attacks characterizes a recognizer's resistance to random guessing while resistance to imitation attacks characterizes a recognizer's resistance to shoulder surfing type of attacks. These features correlate with the EER value of the recognizer; the lower the EER is, the stronger the method will be when distinguishing genuine and imitation gestures.

We note that previous works on free-form gestures for authentication such as Sherman et al. [44] and Yang et al. [53] have provided data on the usability and memorability of these type of gestures in the lab and the field. However, our work is focused on recognition performance in an authentication system, thus, other usability evaluations presented in the previous work are beyond our scope and we refer to Sherman et al. [44] and Yang et al. [53] for details.

#### *Invariance Benchmark: Recognition-Optimal Combination*

$-family gesture recognition schemes such as $1 [51], Protractor [32], $P [48], and $N [2] implement preprocessing steps for removing a gesture's rotation, scale, and location. This is done to minimize the variations in performance of the same gestures by different people and to correspondingly increase recognition accuracy. We define the choice to make these three gesture variables invariant as the *recognition-optimal* combination.

Previous free-form gesture authentication systems based on Protractor [44, 53] used this recognition-optimal combination in their implementations. They demonstrated the effectiveness of distinguishing gestures from different people and great ability to resist shoulder surfing attacks [44]. However, the past work does not show whether this is optimal for authentication purposes. In our analysis, we use the recognition-optimal combination as a benchmark.

### Brute-Force Attack Method

We generated a brute-force attacks with the following steps: 1) Randomly generate two sequences for x and y. 2) Filter the two sequences using a low pass filter with a cutoff frequency at 10 Hz. Remove the first few points that are distorted by the time delay of the filter. 3) Resample the generated gesture by the sampling rate.

The choice for the 10 Hz cutoff is not arbitrary. In our analysis, gestures are resampled to the same length (256). We assume the time to perform a gesture is one second, since we focus on guessing the gesture's shape. By examining the distribution of frequencies for each gesture, we found that the majority of gesture frequencies are concentrated under 10 Hz.

### Imitation Attack Method

We used the imitation attack samples from two public datasets: SUSig [30] and MCTY-100 [37]. The attackers were asked to observe legitimate authentication attempts. They were also asked to practice the attacks as many times they wanted. After the attackers were confident of their imitations, they performed the attacks.

## RECOGNITION ALGORITHMS

We examined and implemented 13 specific algorithms: Protractor, Edit Distance on Real sequence (EDR), Longest Common Subsequences (LCS), Dynamic Time Warping (DTW); Protractor-, EDR-, LCS-, DTW- kernel based SVM; discrete HMM (dHMM), segment HMM (sHMM), continuous HMM (cHMM); and Gaussian Mixture Models. The knowledge obtained with testing and evaluating these algorithms led us to implement two novel multi-expert approaches: SVM-Garda and Garda. We discuss them later in a separate section.

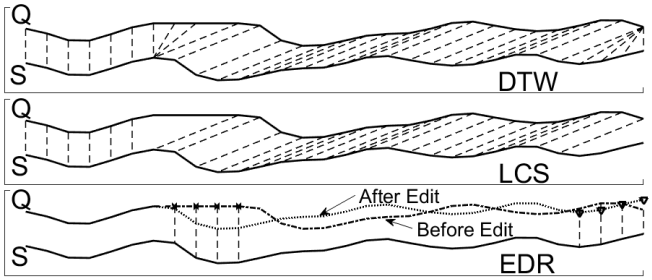


**Figure 3. Examples of three alignment methods: DTW, LCS, and EDR. In DTW, all points in the two sequences must be matched with each other; In LCS, only the same subsequences are matched; In EDR, only the difference between two sequences are counted.**

#### *Protractor Algorithm*

Protractor [32] is a common method used in free-form gesture authentication systems [44, 53]. It is one of the $-family algorithms [51, 32, 48] that uses geometric similarity between two gesture trials. Recall that there are three types of invariance in a gesture: location, scale and rotation invariance. The procedure of Protractor is: resampling a gesture to fixed points, removing the three variants, and finding the maximum value of cosine distances between recall gesture and template gestures.

#### *Time Series Sequence Matching*

Several gesture authentication systems use DTW [40, 4, 18, 52, 5]. LCS [6] and EDR [12] are also similar time series sequence methods. Although LCS and EDR have not been used in published gesture authentication before, they are still alternative methods to DTW and could have higher authentication accuracy. Figure 3 gives examples of these three methods.

Dynamic Time Warping (DTW) is a dynamic programming method for aligning time sequences. Figure 3 shows how DTW seeks the minimum total distance between elements in sequences $Q$ and $S$ by finding the non-decreasing matching pairs of elements. The crucial point is that no elements are discarded through the DTW matching, despite how large the distance between a pair of the elements could be.

Longest Common Subsequences (LCS) [6] is another similarity measurement approach for sequences. The LCS method [6, 34] is used to find the longest common subsequence of two sequences by elastic matching. As shown in the middle of Figure 3, sequence $S$ and $Q$ find the longest subsequence by connecting all small continuous subsequences together. This differs from DTW, which counts the distances of matched sequence points. LCS counts the percentage of matched points in the two sequences – for example, if the length of $S$ and $Q$ is $L_S$ and the number of matched points between the two sequences is $L_m$, the LCS distance is $L_m/L_S$.

Edit Distance on Real sequence (EDR) [12] is a method that measures how different two sequences are by counting the number of inserts, delete, and replace operations that are needed to transform one sequence into the other. As shown in the bottom of Figure 3, the EDR distance of the two sequences $S$ and $Q$ is the number of operations that delete the four points with '$\times$' markers and four insert points with '$\bigtriangledown$'. Unlike LCS or DTW, EDR is a method that does not reward matches but rather penalizes gaps and mismatches.

#### *Support Vector Machine*

Support Vector Machine (SVM) [14] is another technique used in gesture authentication systems [7, 25, 24]. We used a non-linear SVM classification algorithm with a kernel function to transform the input data into a higher-dimensional feature space. The essential property of Protractor and time series sequence matching algorithms (i.e. LCS, DTW, EDR) also measure the similarity between gestures, meaning we can use them as kernel functions for SVM.

#### *Gaussian Mixture Model*

Gaussian Mixture Models (GMM) have been used in voice recognition [10] [11] [38]. It is used to estimate any probability density function. A key feature is that GMM does not consider the order of a sequence. Assume a gesture sequence $X = x_1, ..., x_N$ has D-dimensional feature vectors. The Gaussian Mixture Model of X is a weighted sum of M component Gaussian densities below:

$$p(X|\lambda) = \sum_{i=1}^{M} \omega_i p_i(X)$$

where X is a D-dimensional data vector, $\omega_i, M$, refers to the mixture weight, s and $p_i(X)$ corresponds to the component Gaussian densities. The $p_i(X)$ are parameterized by a mean $D \times 1$ vector $\mu_i$, and a $D \times D$ covariance matrix, $\Sigma_i$:

$$p_i(X) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(X-\mu_i)'(\Sigma_i)^{-1}(X-\mu_i)}$$

The mixture weights, $\omega_i$ satisfy the constraint $\sum_{i=1}^{M} \omega_i = 1$. $\lambda = \{\omega_i, \mu_i, \Sigma_i\}, i = 1, ..., M$.

If there is more than one gesture password, the GMM needs to calculate the universal background model (UBM) with the whole users' data and obtain a single $\lambda_{ubm}$. We then calculate the similarity score between two gestures by doing:

$$Similarity(X_q) = logp(X_q|\lambda_{tmp}) - logp(X_q|\lambda_{ubm})$$

$X_q$ is the recall sequence, $\lambda_{tmp}$ is the template gesture model and $\lambda_{ubm}$ is the UBM model of the dataset.

#### *Hidden Markov Model*

Hidden Markov Model (HMM) is a stochastic model used in many types of gesture recognition systems [31, 21, 19]. There are two main directions to build an HMM for gestures.

One way is using the directions between adjacent gesture points as observations of a HMM and train the model [31]. We call this a discrete HMM (dHMM), wherein we divide the directions into 16 equally-spaced arcs. The number of observations for a single gesture part is therefore 16. We found the seven states left-to-right no jump HMM can output the lowest EER out of many other HMM models. The transition matrix is restricted to what is seen in Figure 4.

The other one is segmenting the gesture to identify the critical points as observations of HMM to develop the HMM model [19]. We label this as segmentation HMM (sHMM), where we segment and classify the gestures into basic parts.

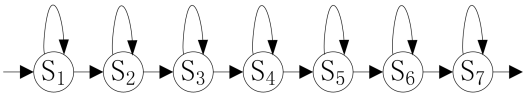


**Figure 4. 7 States Left-to-Right No Jump HMM structure. Every state of the HMM has a self transition loop and only can move forward to next neighbour state.**

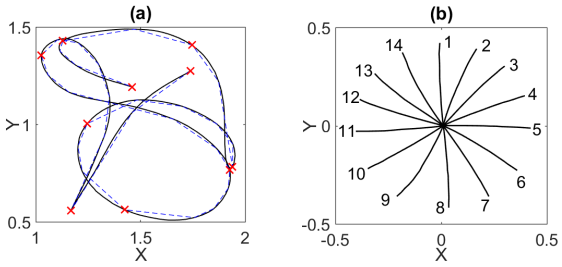


**Figure 5. (a) Example of gesture after RDP algorithm and (b) Clustered basic gesture parts. In figure (a), the solid trajectory is the original gesture, the dashed line is the trajectory of the remaining samples after the first round of RDP, the red cross markers ('×') are the samples after the second round of RDP. We find that the gesture samples after the first RDP is still dense and after a second RDP are ready to be used as splitting points. (b) Basic gestures all start from (0,0) to different directions and can be regarded as 14 observations in HMM.**

Then, we train the HMM based on those basic gesture parts in each gesture.

There are three steps to build the sHMM. 1) Segmentation; we use the Ramer-Douglas-Peucker (RDP) [26] algorithm for two rounds to reduce the number of points in the gesture. RDP is an approximation method to find a similar curve to the original sequence using fewer points. The dashed line and '×' in Figure 5(a) shows the approximation gesture after the first and second round of RDP, respectively. With the approximation points '×', we segmented the gestures to several basic gesture parts. 2) Clustering. We cluster the segmented gestures into a few basic parts. We first normalize the gesture segments to the same size, then implement K-means as an unsupervised learning method to classify the basic gesture parts into several classes. We found, through observation, that K=14 was the optimal number for basic gesture types. Figure 5(b) shows the 14 basic gesture parts. Thus, the gestures can be represented as combinations of these 14 basic gesture parts. 3) Train HMM. We follow the steps in dHMM to build the HMM for each gesture password from these 14 parts.

In addition to the above, we implemented a continuous HMM (cHMM). Specifically, we divided a gesture evenly into $N$ basic gesture parts. For each basic gesture part, there are several gesture points – we modeled the probability density function of these points with GMMs. Each basic gesture part can be represented by a combination of $M$ Gaussian distributions. By training the HMM with the Baum-Welch algorithm [29] we obtain HMM verifiers for each gesture type. The Baum-Welch algorithm is a special case of the Expectation-Maximization (EM) algorithm [3] and is used to find a locally optimal solution to the HMM training problem.

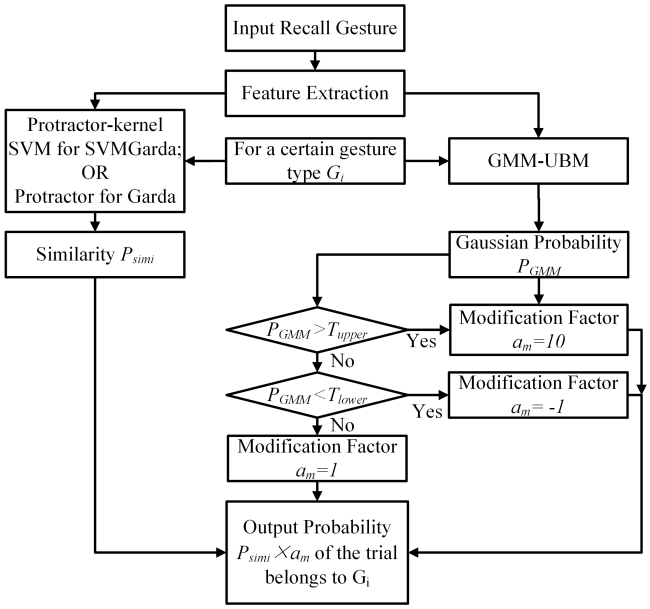


**Figure 6. SVMGarda and Garda authentication system. SVMGarda system uses the Protractor kernel SVM method in the top left part of authentication section. Garda system uses the Protractor method. The similarity of Garda between the recall and template gesture is measured by Protractor and GMM-UBM separately. The similarity output of Protractor is $P_{simi}$. It is modified by different factors based on the comparison result of Gaussian probability of GMM-UBM $P_{GMM}$, and two thresholds $T_{upper}$ and $T_{lower}$. If $P_{GMM} > T_{upper}$, it means that GMM-UBM recognizer is confident that two gestures are similar, so the final similarity score of Garda is $10 \times P_{simi}$. If $P_{GMM} < T_{lower}$, it means the GMM-UBM recognizer is confident that two gestures are not similar at all and the final result is $-P_{simi}$. Otherwise, GMM-UBM cannot make a confident judgment, so the final similarity of Garda is the same as $P_{simi}$.**

## GARDA AND SVMGARDA AUTHENTICATION SYSTEMS

In this section, we describe our novel gesture authentication system. We developed two different types of multi-expert systems. The first one, called Garda, combines Protractor and GMM-UBM. The other one, called SVMGarda, combines the Protractor kernel SVM and GMM-UBM. Figure 6 gives an overview of our Garda and SVMGarda authentication systems. Unlike other recognizers, which are based on one dimension of recognition, Garda and SVMGarda are better because they recognize gestures in two dimensions: the gesture shape feature (by GMM-UBM) and adjacent points feature (by Protractor or by Protractor kernel SVM).

Multi-expert (ME) classification systems work by combining the results of different classifiers to make the final classification decision [39]. Multi-expert systems can work better than other recognizers since they measure gesture similarities using multiple recognizers. Since the similarity results from the different recognizers are not correlated, the multi-expert system can combine results to achieve a lower EER. We define the similarity score from one recognizer as a single dimension of recognition.

The rationale for selecting Protractor instead of the other time sequence matching method in Garda is that the Protractor has a lower EER compared to LCS, EDR, and DTW. The rationale for combining the GMM-UBM and Protractor is that the

| Dataset | Recall Set (Trial #) | Gesture (Types #) | Screen Size (Inches) | Samples |
|---|---|---|---|---|
| **Freeform (Set 1)** | 11-12 | 56 | 10.1 |  |
| **Freeform (Set 2)** | 13-17 | 54 | 10.1 |  |
| **$1 Demo** | 11-30 | 16 | 3.8 |  |
| **Vatavu (Set 1)** | 11-20 | 18 | 6.1 |  |
| **Vatavu (Set 2)** | 11-20 | 20 | 6.1 |  |
| **MMG corput** | 11-30 | 16 | 13.3 |  |
| **HHReco** | 11-30 | 13 | 6.2 |  |
| **SUSig** | 11-30 | 94 | 3.7 |  |
| **MCYT -100** | 11-50 | 100 | 6.3 |  |

**Table 1. Summary of the seven datasets used in our analysis. The template set of each dataset is always the first ten trials (# 1 to #10) of every type of gesture. The screen size refers to the screen size of device on which the gesture samples were collected for that dataset. SUSig and MCYT-100 datasets also included attacks, which were used in our attack evaluations.**

two methods focus on different features of gestures. GMM-UBM does not take the order of gesture points into account, which means it only focuses on the shape of a gesture. Protractor concentrates on the trajectory of a gesture, especially the connections between adjacent gesture points.

There are three steps in both systems. First, the recall gesture is sent to the GMM-UBM authentication verifier. We set two thresholds $T_{upper}$ and $T_{lower}$ for the Gaussian Probability $P_{GMM}$. If $P_{GMM} > T_{upper}$, we can make the decision that this recall gesture belongs to a certain gesture type $G_i$. If $P_{GMM} < T_{lower}$, we can decide that this gesture does not belong to gesture type $G_i$. Otherwise, we cannot decide the gesture's type. Second, the recall gesture is sent to a time-series authentication recognizer. For the Garda system, we use Protractor to measure the similarity. For SVMGarda system, we use the Protractor kernel SVM to measure the similarity $P_{simi}$ that a recall gesture belongs to a certain gesture type $G_i$. In the last step, we combine the two probabilities together by modifying $P_{simi}$ with a modification factor $a_m$, which is determined by the GMM-UBM verifier.

## DATASETS

Table 1 shows a summary of the datasets we used in our analysis: (i) Freeform gesture dataset [44], (ii) $1 Demo dataset [51], (iii) Vatavu's gesture datasets [49], (iv) MMG corpus dataset [1], (v) HHReco dataset [27], (vi) SUSig [30], and (vii) MCTY-100 [37]. For consistency, we opted to use the first 10 trials from each dataset as Template sets. The remaining trials are called the Recall set. We use the first five datasets to test the authentication accuracy of different recognizers. We use the last two datasets to examine the ability of the recognizers to resist imitation attacks. In total, we used 10872 gestures collected from 328 participants across these seven datasets.

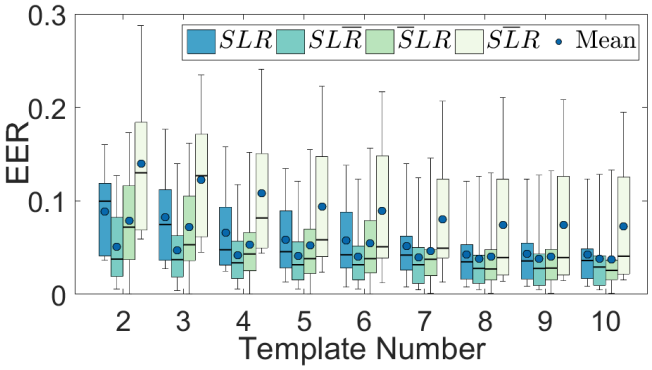


**Figure 7. EER values under combinations ($SLR$), ($SL\bar{R}$), ($\bar{S}LR$), and ($S\bar{L}R$). $S$, $L$, and $R$ mean the gesture's scale, location, are rotation are invariants. $\bar{S}$, $\bar{L}$, and $\bar{R}$ mean the these three are variants. The whiskers show the maximum and minimum EER of each group. We can see that rotation alone as variant ($SL\bar{R}$) has obvious positive effect authentication accuracy; scale alone as variant ($\bar{S}LR$) has no obvious effect authentication accuracy; location alone as variant ($S\bar{L}R$) has obvious negative effect on the authentication accuracy.**

## RESULTS

We first present the results of our analysis of the invariances (rotation, scale, location) and their effect on authentication performance. We follow up with results on the performance of the different recognizers and present results of the brute-force and imitation attacks. Finally, we present the implementation and evaluation of our novel approach Garda operating on a mobile device.

We used two-way ANOVA for testing the statistical significance between the authentication performances of individual gesture invariances and the benchmark since both the invariance combination types and the template numbers can affect the authentication result. We used repeated measures one-way ANOVA to test the statistical significance between the authentication performances of the 13 recognizers on different datasets. We use Bonferroni corrected p-values for the post-hoc test for controlling the familywise error.

### Invariances Analysis

Figure 7 shows the individual effects of rotation, location and scaling on authentication performance. In this analysis, we used the EER of the recognition-optimal combination as the benchmark.

A two-way ANOVA test indicated a statistically significant difference between individual gesture invariance authentication performance and the benchmark ($\chi^2(251) = 11.6$, $p < 0.001$), while the interaction effect between the two factors is not statistically significant ($\chi^2(251) = 0.17$, $p = 1$). Thus, we did not consider the interaction effect between the the invariance combination types and the template numbers.

#### *Rotation Invariance*

Figure 7 demonstrates that making the gesture rotation as variant can improve the recognition performance (i.e. reduce the EER). The result holds true even with a different number of template gestures across different gesture datasets. This can be explained by a users' drawing habit – people will tend to input their gesture into the tablet the same way every time.

Very rarely, however, do gestures in the same dataset have the same rotation angle as other gestures.

The statistical test between $SL\overline{R}$ and $SLR$ with template number (2 to 10) shows there is no statistically significant difference ($p = 0.373$, $d = 0.42$). The lack of significance stems from the fact that the EER values become relatively close as the number of templates becomes larger (8, 9, 10). A recognizer is more likely to identify a genuine trial when there are more genuine templates to test against. This reduces, overall, the effect of invariance combinations on EER. If we do not consider EERs with large template number (8, 9, 10), then there is statistically significant difference between $SL\overline{R}$ and $SLR$ ($p = 0.013$, $d = 0.56$). As such, allowing for rotation variance can reduce the possibility of accepting false gestures.

*Scale Invariance*

Figure 7 shows that the scale variable has a slight, positive influence on recognition performance. The primary reason is that Protractor is based on the cosine distance between two gesture sequences. The cosine distance measures the directional difference between two vectors and ignores the Euclidean distance. Thus, the difference among the size of how the gestures are drawn will not affect the relative similarity scores. The statistical test between $\overline{S}LR$ and $SLR$ shows no statistically significant difference ($p = 1$, $d = 0.14$). Taking scale as variant does not have a statistically significant influence on authentication performance.

*Location Invariance*

Figure 7 shows that taking location as variant has a negative influence on recognition accuracy. The primary reason is that it is hard for people to draw at the same location on the screen when repeating their gestures. The recognition performance is not based on the absolute similarity among trials of the same person; it depends on the relative difference of similarities between genuine trials and imitations. Taking gesture location as variant will reduce the similarity of genuine gestures and increase the similarity of imitation gestures. This makes it more difficult to distinguish the genuine and imitation gestures, and the authentication performance will be worse. The statistical test between $S\overline{L}R$ and $SLR$ shows a statistically significant difference ($p = 9.788 \times 10^{-4}$, $d = 0.61$).

*Variants Combinations Effects on Recognition*

Figure 8 shows that the combination ($\overline{S}L\overline{R}$) achieves the lowest EER (0.041) on average. It can be explained by the individual effects of the three variables: rotation variant has statistically significant positive effect on EER, location variant has statistically significant negative effect on EER, and scale variant has slightly positive effect on the EER. Compared to the common case of the three combinations ($SLR$), where the EER = 0.075, the $\overline{S}L\overline{R}$ reduces the EER by 45.3%.

Additionally, the ROC curves of $\overline{S}L\overline{R}$ and $SL\overline{R}$ are very close. It verifies our analysis that the gesture scale variable has no statistically significant effect on EER.

We also can observe that the four combinations that take location as variant ($S\overline{L}R$, $S\overline{L}\overline{R}$, $\overline{S}\overline{L}R$, and $\overline{S}\overline{L}\overline{R}$) are the lowest four ROC curves. This means that, with the same conditions for gesture scale and rotation, taking gesture location as variant

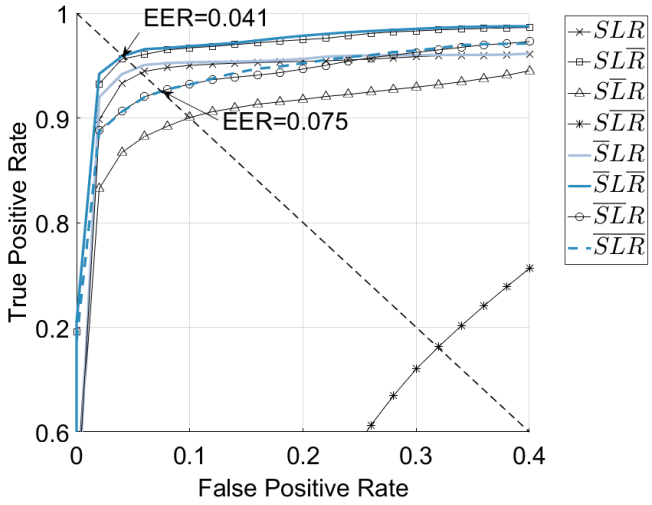


**Figure 8. The average ROC curves of the five gesture datasets with two to ten template gestures for the eight combinations of the three variants (Scale, Location, Rotation). $S$, $L$, and $R$ mean the gesture's scale, location, are rotation are invariants. $\overline{S}$, $\overline{L}$, and $\overline{R}$ mean the these are variant. We can see that the combination ($\overline{S}L\overline{R}$) has the lowest EER (=0.041) across the five datasets. While the recognition-optimal case ($SLR$) can only achieve EER=0.075. We conclude that combination ($\overline{S}L\overline{R}$) is the optimal authentication selection of three variants.**

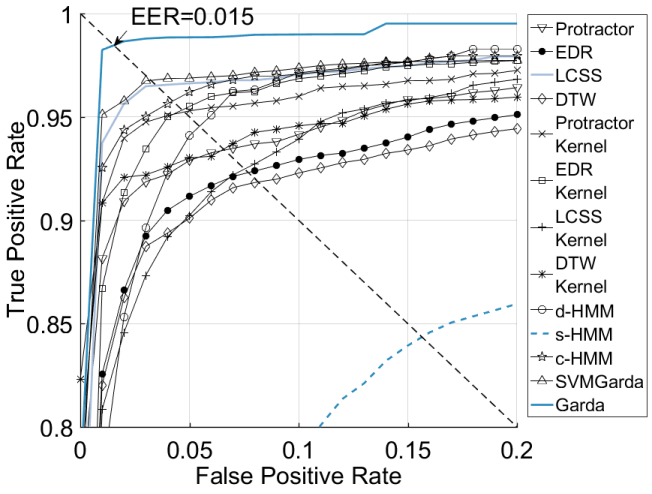


**Figure 9. The average ROC curves for the 13 recognition methods over the five gesture datasets. We found that Garda has the lowest EER (0.015). Since the ROC curve of Garda is closest to the up-left corner, it should be the most tolerant of the change of authentication threshold. We conclude that Garda is the best among the 13 methods.**

will always have a negative effect on authentication accuracy. Generally, the reason is that people cannot keep their gesture's location at a relative fixed place on the touch screen. This means that different gesture locations can distort the similarities among the genuine and fake gestures.

### Performance of Different Recognition Methods

To select the optimal recognition method for gestures, we tried 13 different methods in four groups: sequence matching group (Protractor, EDR, LCS, DTW), One-class SVM group (Protractor-, EDR-,LCS-,DTW-kernels), HMM group (dHMM, sHMM, cHMM), and multi-expert group (SVM-

| | Freeform (Set 1) | | Freeform (Set 2) | | MMG | | $1 Demo | | HHReco | | Vatavu (Set 1) | | Vatavu (Set 2) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | EER | t (ms) | EER | t (ms) | EER | t (ms) | EER | t (ms) | EER | t (ms) | EER | t (ms) | EER | t (ms) |
| **Protractor** | 0.027 | 0.7 | 0.105 | 0.5 | 0.032 | 0.6 | 0.021 | 0.6 | 0.015 | 0.7 | 0.008 | 0.8 | 0.005 | 0.7 |
| **EDR** | 0.062 | 27.9 | 0.137 | 27.0 | 0.185 | 42.8 | 0.044 | 42.4 | 0.032 | 43.6 | 0.031 | 43.1 | 0.010 | 41.2 |
| **LCS** | 0.036 | 22.2 | 0.121 | 21.2 | 0.056 | 34.1 | 0.020 | 33.8 | 0.013 | 34.9 | 0.011 | 35.1 | 0.002 | 34.9 |
| **DTW** | 0.062 | 3.9 | **0.142** | 3.5 | **0.204** | 5.5 | 0.049 | 5.5 | 0.026 | 5.5 | 0.011 | 36.8 | 0.002 | 36.5 |
| **Protractor Kernel** | 0.035 | 5.1 | 0.141 | 4.8 | 0.046 | 7.2 | 0.032 | 5.6 | 0.047 | 6.0 | 0.004 | 6.7 | 0.011 | 4.6 |
| **EDR Kernel** | 0.048 | 391.1 | 0.055 | 419.2 | 0.119 | 464.1 | 0.034 | 112.7 | 0.037 | 128.5 | 0.009 | 101.2 | 0.003 | 90.4 |
| **LCS Kernel** | 0.036 | 368.3 | 0.118 | 393.9 | 0.141 | 489.2 | 0.058 | 149.0 | 0.040 | 162.6 | 0.006 | 141.5 | 0.006 | 121.1 |
| **DTW Kernel** | 0.053 | 471.5 | 0.113 | 449.2 | 0.063 | 641.1 | 0.092 | 314.5 | 0.028 | 333.2 | 0.001 | 334.1 | 0.013 | 339.1 |
| **d-HMM** | 0.036 | 1.5 | .113 | 1.5 | 0.065 | 1.4 | 0.045 | 1.4 | 0.038 | 1.5 | 0.023 | 2.0 | 0.011 | 2.1 |
| **s-HMM** | **0.109** | 1.6 | 0.130 | 1.2 | 0.198 | 1.3 | **0.165** | 1.3 | **0.144** | 1.6 | **0.140** | 1.9 | **0.104** | 1.8 |
| **c-HMM** | 0.044 | 3.7 | 0.104 | 3.1 | 0.052 | 4.9 | 0.015 | 4.4 | 0.034 | 4.9 | 0.005 | 5.7 | 0 | 5.7 |
| **SVMGarda** | 0.026 | 5.3 | 0.116 | 5.2 | ***0.032*** | 6.3 | 0.014 | 5.7 | 0.012 | 5.6 | 0.011 | 7.1 | 0 | 4.7 |
| **Garda** | ***0.019*** | 1.0 | ***0.047*** | 2.4 | 0.033 | 2.1 | ***0.012*** | 1.7 | ***0.007*** | 1.8 | ***0*** | 2.2 | ***0*** | 2.1 |

**Table 2. EER values against estimated authentication time. Each recognition method is implemented in MATLAB and tested on five gesture datasets in terms of EER and authentication time (t) in milliseconds. Since the authentication time is based on MATLAB computations, it can be only used for a relative comparison among different recognizers. In each dataset, the lowest EER is shown in bold and italic, while the highest EER is only bold. Generally, the ME group always has the lowest EER among different datasets and authentication methods. Between the two ME methods, their EER performances are dependent on the different datasets. However, since Garda has much lower EER than SVMGarda in Freeform (set 2) and the computation cost of Garda is always lower than SVMGarda, we conclude that Garda is the best among the 13 methods.**

Garda, Garda). Since our analysis of gesture invariances is based on Protractor, we also used Protractor as the baseline for the comparison of these 13 methods. We evaluated the above methods with EER values, authentication times, brute-force and forgery attacks.

We used 10 gesture trials as the template gesture set to minimize the bias from the template selection. The reason is that the selection of template gestures may effect the recognition performance. The more gesture templates, the less bias that exists in the selected templates.

Figure 9 shows that Garda is the best among the 13 recognizers. We averaged the ROC curves of the five gesture datasets for the 13 recognizers. We found that Garda achieves lowest EER (0.015) in the averaged ROC. The ROC curve of Garda is the closest to the up-left corner, meaning that Garda can recognize the most genuine gestures correctly while rejecting attacks as well as other recognizers.

Table 2 shows that Garda has most of the lowest EERs among the 13 recognition methods through the five datasets. The authentication time for Garda is around 2 ms, which is also among the lowest authentication times. Specifically, both of the two multi-expert methods, Garda and SVMGarda outperform the other recognizers. Garda is also more stable than SVMGarda. For example, in Freeform (Set 2), Garda achieved the lowest EER (0.047), while SVMGarda got a relatively much higher EER (0.116).

Based on repeated measures ANOVA analysis on the 13 recognizers, there is a statistically significant difference between EERs when choosing different recognizers ($\chi^2(90) = 9.71$, $p < 0.001$). From our post-hoc analysis, we found there are no statistically significant differences between Garda and Protractor ($p = 0.121$, $d = 0.52$), Garda and LCS ($p = 0.0749$, $d = 0.68$), Garda and DTW ($p = 0.0545$, $d = 1.17$), Garda and Protractor kernel SVM ($p = 0.0528$, $d = 0.89$), and Garda and SVMGarda ($p = 0.208$, $d = 0.47$). This does not impact our result, since Garda is still the best performing authenticator for the following reasons: First, Protractor cannot prevent brute-force and imitation attacks as well as Garda irrespective of whether its EER is statistically significantly different; Second, Figure 9 also shows that the ROC curve of Garda is much more ideal than the other recognizers.

We found that the authentication times of the one-class SVM group are much longer than the sequence matching group while the EER values are relatively similar. The reason for the longer authentication time is that, with the similarity matrix, SVM methods need an extra step to examine the similarity scores with training gesture trials and converting the similarity to probabilities. Since one-class SVM group distinguishes the gestures based on the sequence matching kernel functions, its ability to distinguish gestures should be the same as sequence matching group methods. Thus, the one-class SVM group has similar EER as sequence matching group methods.

The sHMM method always performs with the worst EER. The reason is that we segment the gesture into several parts based on sharp turns and then classify those parts into 14 observations. We tend to lose a lot of useful information and this makes the gestures more likely to be misclassified.

#### *Performance Under Brute-Force Attacks*

Figure 10 shows how the authentication methods resist brute-force attacks. We found that the recognizers' performance are polarized. On the one hand, EDR, LCS, DTW, dHMM, sHMM and Garda resisted all of the brute-force attacks. On the other hand, the brute-force attack cracked most of the gestures of the other recognizers. Specifically, we found that the features of recognizers that can resist attacks are distance based (such as EDR, LCS, and DTW) and time series HMM-based (dHMM, sHMM). SVM was generally weak against these attacks.

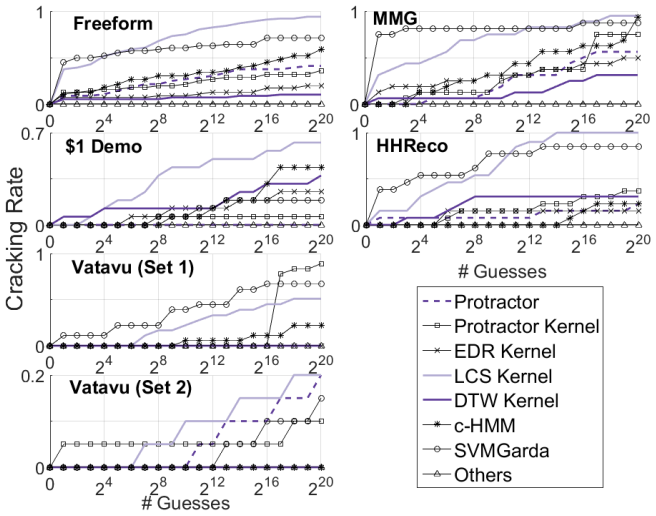


**Figure 10. Guessing success rates of brute-force attacks of the 13 authentication systems on different datasets. The "Others" includes Garda, EDR, LCS, DTW, dHMM, and cHMM. From the cracking rates, Garda, EDR, LCS, DTW, dHMM, and cHMM have the best ability on resisting brute force attacks since the success cracking rates of them are 0% through the 6 datasets.**

| Method | MCYT-100 | SUSig |
|---|---|---|
| **Protractor** | 0.151 | 0.350 |
| **EDR** | 0.155 | 0.366 |
| **LCS** | 0.101 | 0.420 |
| **DTW** | 0.189 | 0.371 |
| **Protractor Kernel** | 0.163 | 0.335 |
| **EDR Kernel** | 0.103 | 0.402 |
| **LCS Kernel** | 0.084 | 0.344 |
| **DTW Kernel** | 0.093 | 0.397 |
| **d-HMM** | 0.233 | 0.333 |
| **s-HMM** | 0.338 | 0.445 |
| **c-HMM** | 0.125 | 0.202 |
| **SVMGarda** | 0.096 | 0.479 |
| **Garda** | 0.045 | 0.035 |

**Table 3. The EER of the 13 authentication methods in MCYT-100 and SUSig datasets under skilled forgery attacks. Only Garda has considerable advantages in EER.**

### *Performance Under Imitation Attacks*

We used the datasets MCYT-100 and SUSig to examine the 13 methods' performance under imitation attacks. These datasets have samples from both legitimate users and imitation attacks from skilled attackers. We used the first ten legitimate trials for each user as templates and the rest as legitimate authentication attempts. The skilled attackers observed the genuine trials and practiced them until they felt confident with attacks.

Table 3 shows that only Garda outperforms the rest 12 methods on resisting forgery attacks. EER values of Garda are 0.045 and 0.035 in MCYT-100 and SUSig, respectively. In contrast, the lowest EER values for the other methods are 0.084 (LCS Kernel) and 0.202 (c-HMM). In summary, Garda outperforms all of the other approaches against imitation attacks, and none of the other approaches perform well against both of the datasets.

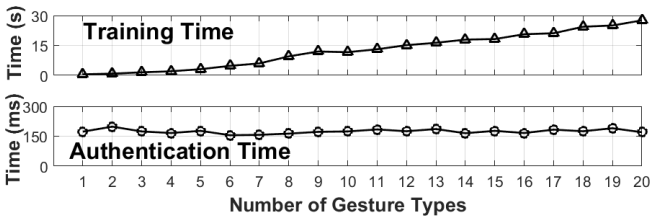


**Figure 11. The processing time for training and authentication under different number of gesture passwords. The upper figure shows the time for training, the lower shows the time for authentication. Along with the increasing number of gesture passwords, the training time is gradually increasing while the authentication time stays stable.**

### Mobile Device Implementation and Evaluation

We implemented Garda, described in Figure 6, on an Android platform. Our test device was a Samsung Galaxy Note 10.1, which had a 1.9 GHz Quad CPU and 3 GB of RAM. For our evaluation, we created 20 new gesture passwords (two templates per gesture) on the mobile device and recorded the processing time for training and authentication with a different number of gesture passwords.

Figure 11 shows our mobile device evaluation results. With the number of gestures (users) increasing, we see that the training time increases while the authentication time stabilizes around 150 ms. The training time increases because Garda uses the Expectation-Maximum algorithm to train the UBM based on all the gesture trials in the dataset and re-train the Gaussian Mixture Model (GMM) for all gesture types based on this new UBM. More gestures lead to more time spent training models. In the authentication phase, we only need one GMM for a given gesture so the authentication time remains stable. However, there are typically not multiple passwords on a mobile device, and the training time even with 20 different gestures is only 30 seconds and could be run in-background over other tasks. All this shows that Garda is an efficient gesture recognizer for a mobile device authentication system.

## DISCUSSION

Our evaluation showed that our novel gesture authentication method - Garda - had the best authentication EER (0.015), shortest authentication time, and resisted all of the brute-force attacks and most of the forgery attacks (EER = 0.040). Garda achieves this by combining the individual advantages of the Protractor and GMM-UBM recognition schemes. Protractor identifies the gestures temporally while GMM-UBM identifies the temporal-independent distributions of the coordinate points. Combining these two methods resulted in a more harmonious, effective authentication system.

SVMGarda performed second-best to Garda. The key difference between the two is that SVMGarda uses the cosine similarity from Protractor as a kernel function for the support vector classifier. Protractor uses inverse cosine distance – the least similar gestures are scored at '0' and the most similar gestures (exact match) are scored as infinite. When Protractor is passed through the SVM as a kernel function, the output is changed from a gesture score ranging from [0,inf] to a similarity probability ranging from [0,1]. This has the effect of

shrinking the difference between genuine and imitation gestures, making it more likely that SVMGarda will mis-classify.

sHMM performs the worst with an EER of 0.157. Following up gesture segmentation with a classification of the segmented parts removed crucial details from the genuine gestures. This had the side-effect of making them harder to distinguish. Of course, this is a function of the number of gesture segments: if we increase the segments, we can preserve more detail. If we continue to increase segments, sHMM becomes the same as dHMM. This improves the EER from 0.157 to 0.056.

Six recognizers (EDR, LCS, DTW, dHMM, sHMM, and Garda) were capable of resisting all brute-force attacks, but for different reasons. EDR, LCS, and DTW use recognition methods based on the Euclidean distance between genuine and fake gestures. If the gesture points are randomly generated, as they are in the brute-force method, then the matching problem is almost impossible: a randomly generated collection of a hundred-or-so coordinate points need to fall close to a continuous sequence of gesture points clustered very closely together out of thousands of positions on the screen. The result is not surprising when it is viewed this way. For dHMM and sHMM: the randomly generated gesture contains many sudden, random turns that can be used as segmentation points when compared to a user-generated gesture.

There are seven recognizers that were unable to resist brute-force attacks. Authentication thresholds in each scheme are trained to distinguish different users' gestures, and these thresholds are quite loose since they only need to recognize a single genuine gesture from entirely different gestures. There are three reasons for these three groups: Protractor-based methods (Protractor, Protractor-Kernel SVM, SVMGarda), SVM-based methods (Protractor, LCS, DTW, and EDR kernel SVM and SVMGarda), and cHMM. For Protractor-based methods: it depends on cosine distance, which only cares about the directions between two adjacent points. This is easier than matching gestures based on Euclidean distance. For SVM-based methods, it is the same reason as why the EER values are so high: the method transforms a large range of scores [0,inf] to a limited range of probabilities [0,1], increasing the opportunity for mis-classification. cHMM used the distribution of gesture points locations without considering the temporal order, making the brute-force attack more likely to crack cHMM since the attacks are based on randomly generated gestures.

For the imitation attacks, Garda achieved the lowest EER out of all the other methods. The reason that other methods cannot resist imitation attacks is that single time sequence authentication methods are unable to distinguish the variations in genuine gestures and the difference between the genuine and skilled attack gestures.

We can see that the Garda method is robust and practical for mobile authentication. First, the authentication time is short (200 ms) and is also relatively stable as the number of gestures increases. Although we tested 20 gestures (20 different users) in the system to judge its viability, we note that it is unlikely a personal device would have so many different accounts. Even with 20 gestures, 30 seconds is affordable for a one-time training process. The mobile device only needs to store the trials, GMM, and UBM models; therefore, our system can have a short and stable authentication time and still be scalable for large numbers of gestures.

To best of our knowledge, we have presented the first analysis of how the invariances of gesture preprocessing impact the performance of gesture-based authentication systems. We found that an optimal combination (scale variant, location invariant, rotation variant) achieves the lowest EER (0.041) on average. This derives from the combination of individual effects of the three variables: rotation variant has a statistically significant, positive effect on EER; location variant has a statistically significant negative effect on EER; scale variant does not have a statistically significant effect on the EER. The screen sizes on which gestures were collected, 3.75in to 13.3 inches, do not appear to affect our analysis of gesture invariance.

The gesture variable combinations with EERs higher than $\overline{S}L\overline{R}$ can be explained by the individual effects of the three variables. First, since people are unable to draw their gestures at the same place every time, taking gesture's location as variant always has a negative effect on authentication accuracy. Second, these three variables interact with each other. The binary state of one variable can influence other variables, and thus, influence the final recognition accuracy.

## CONCLUSIONS

We presented and evaluated a novel multi-expert gesture recognizer design for authentication: Garda. We also implemented and evaluated Garda on a mobile device. All our results show that our implementation can largely improve the performance of gesture-based authentication systems. Garda was the final result of a rigorous evaluation of 13 different methods to implement gesture recognizers. We applied several datasets and two different attacks against the recognizers. Finally, we conducted the first analysis of how tuning the variables of preprocessing methods of gesture recognizers can impact their authentication performance. The presented authentication-optimal combination can reduce up to 45.3% of EER on average compared to recognition-optimal configuration used in previous work. Additional material available at `http://securegestures.org`.

## ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Number 1228777. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Gradeigh D. Clark was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

## REFERENCES

1. Lisa Anthony, Radu-Daniel Vatavu, and Jacob O. Wobbrock. 2013. Understanding the Consistency of Users' Pen and Finger Stroke Gesture Articulation. In Proceedings of Graphics Interface 2013 (GI '13). Canadian Information Processing Society, Toronto, Ont.,

Canada, Canada, 87–94. http://dl.acm.org/citation.cfm?id=2532129.2532145

2. Lisa Anthony and Jacob O. Wobbrock. 2010. A Lightweight Multistroke Recognizer for User Interface Prototypes. In Proceedings of Graphics Interface 2010 (GI '10). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 245–252. http://dl.acm.org/citation.cfm?id=1839214.1839258

3. Donald B. Rubin Arthur P. Dempster, Nan M. Laird. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39, 1 (1977), 1–38. http://www.jstor.org/stable/2984875

4. Ilhan Aslan, Andreas Uhl, Alexander Meschtscherjakov, and Manfred Tscheligi. 2014. Mid-air Authentication Gestures: An Exploration of Authentication Based on Palm and Finger Motions. In Proceedings of the 16th International Conference on Multimodal Interaction (ICMI '14). ACM, New York, NY, USA, 311–318. DOI: http://dx.doi.org/10.1145/2663204.2663246

5. Md Tanvir Islam Aumi and Sven Kratz. 2014. AirAuth: Evaluating In-air Hand Gestures for Authentication. In Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14). ACM, New York, NY, USA, 309–318. DOI: http://dx.doi.org/10.1145/2628363.2628388

6. Lasse Bergroth., Harri Hakonen., and Timo Raita. 2000. A Survey of Longest Common Subsequence Algorithms. In Proceedings of the Seventh International Symposium on String Processing Information Retrieval (SPIRE'00) (SPIRE '00). IEEE Computer Society, Washington, DC, USA. DOI: http://dx.doi.org/10.1109/SPIRE.2000.878178

7. 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 (MobiCom '13). ACM, New York, NY, USA, 187–190. DOI: http://dx.doi.org/10.1145/2500423.2504572

8. Joseph Bonneau. 2012. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In 2012 IEEE Symposium on Security and Privacy. 538–552. DOI: http://dx.doi.org/10.1109/SP.2012.49

9. Ulrich Burgbacher and Klaus Hinrichs. 2014. An Implicit Author Verification System for Text Messages Based on Gesture Typing Biometrics. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). ACM, New York, NY, USA, 2951–2954. DOI: http://dx.doi.org/10.1145/2556288.2557346

10. William M Campbell, Joseph P Campbell, Douglas A Reynolds, Elliot Singer, and Pedro A Torres-Carrasquillo. 2006a. Support Vector Machines for Speaker and Language Recognition. Computer Speech & Language 20, 2 (2006), 210–229. DOI: http://dx.doi.org/10.1016/j.csl.2005.06.003

11. William M Campbell, Douglas E Sturim, and Douglas A Reynolds. 2006b. Support Vector Machines Using GMM Supervectors for Speaker Verification. Signal Processing Letters, IEEE 13, 5 (2006), 308–311. DOI: http://dx.doi.org/10.1109/LSP.2006.870086

12. Lei Chen, M. Tamer Özsu, and Vincent Oria. 2005. Robust and Fast Similarity Search for Moving Object Trajectories. In Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD '05). ACM, New York, NY, USA, 491–502. DOI: http://dx.doi.org/10.1145/1066157.1066213

13. Gradeigh D. Clark and Janne Lindqvist. 2015. Engineering Gesture-Based Authentication Systems. Pervasive Computing, IEEE 14, 1 (Jan 2015), 18–25. DOI: http://dx.doi.org/10.1109/MPRV.2015.6

14. Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. Machine Learning 20, 3 (1995), 273–297. DOI: http://dx.doi.org/10.1007/BF00994018

15. Jacek Czyz, Mohammad Sadeghi, Josef Kittler, and Luc Vandendorpe. 2004. Decision fusion for face authentication. In Biometric Authentication. Springer, 686–693. DOI: http://dx.doi.org/10.1007/978-3-540-25948-0_93

16. Alexander De Luca, Alina Hang, Frederik Brudy, Christian Lindner, and Heinrich Hussmann. 2012. Touch Me Once and I Know It's You!: Implicit Authentication Based on Touch Screen Patterns. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 987–996. DOI: http://dx.doi.org/10.1145/2207676.2208544

17. Alexander De Luca, Marian Harbach, Emanuel von Zezschwitz, Max-Emanuel Maurer, Bernhard Ewald Slawik, Heinrich Hussmann, and Matthew Smith. 2014. Now You See Me, Now You Don't: Protecting Smartphone Authentication from Shoulder Surfers. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). ACM, New York, NY, USA, 2937–2946. DOI: http://dx.doi.org/10.1145/2556288.2557097

18. Alexander De Luca, Emanuel von Zezschwitz, Ngo Dieu Huong Nguyen, Max-Emanuel Maurer, Elisa Rubegni, Marcello Paolo Scipioni, and Marc Langheinrich. 2013. Back-of-device Authentication on Smartphones. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, 2389–2398. DOI: http://dx.doi.org/10.1145/2470654.2481330

19. Jiang-Wen Deng and Hung tat Tsui. 2000. An HMM-based Approach for Gesture Segmentation and Recognition. In Pattern Recognition, 2000. Proceedings. 15th International Conference on, Vol. 3. 679–682 vol.3. DOI: http://dx.doi.org/10.1109/ICPR.2000.903636

20. Giovanni Dimauro, Sebastiano Impedovo, Giuseppe Pirlo, and A Salzo. 1997. A multi-expert signature verification system for bankcheck processing. International Journal of Pattern Recognition and Artificial Intelligence 11, 05 (1997), 827–844. DOI: http://dx.doi.org/10.1142/S0218001497000378

21. Mahmoud Elmezain, Ayoub Al-Hamadi, and Bernd Michaelis. 2009. Hand Trajectory-based Gesture Spotting and Recognition Using HMM. In 2009 16th IEEE International Conference on Image Processing (ICIP). 3577–3580. DOI: http://dx.doi.org/10.1109/ICIP.2009.5414322

22. Tom Fawcett. 2006. An Introduction to ROC Analysis. Pattern Recogn. Lett. 27, 8 (June 2006), 861–874. DOI: http://dx.doi.org/10.1016/j.patrec.2005.10.010

23. Dinei Florencio and Cormac Herley. 2007. A Large-scale Study of Web Password Habits. In Proceedings of the 16th International Conference on World Wide Web (WWW '07). ACM, New York, NY, USA, 657–666. DOI: http://dx.doi.org/10.1145/1242572.1242661

24. Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. 2013. Touchalytics: On the Applicability of Touchscreen Input As a Behavioral Biometric for Continuous Authentication. Trans. Info. For. Sec. 8, 1 (Jan. 2013), 136–148. DOI: http://dx.doi.org/10.1109/TIFS.2012.2225048

25. Eiji Hayashi, Manuel Maas, and Jason I. Hong. 2014. Wave to Me: User Identification Using Body Lengths and Natural Gestures. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14). ACM, New York, NY, USA, 3453–3462. DOI: http://dx.doi.org/10.1145/2556288.2557043

26. Paul S Heckbert and Michael Garland. 1997. Survey of Polygonal Surface Simplification Algorithms. Technical Report. DTIC Document. http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA461098

27. Heloise Hse and A. Richard Newton. 2004. Sketched Symbol Recognition Using Zernike Moments. In Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1 - Volume 01 (ICPR '04). IEEE Computer Society, Washington, DC, USA, 367–370. DOI: http://dx.doi.org/10.1109/ICPR.2004.838

28. Anil K. Jain, Arun Ross, and Sharath Pankanti. 2006. Biometrics: A Tool for Information Security. Trans. Info. For. Sec. 1, 2 (Nov. 2006), 125–143. DOI: http://dx.doi.org/10.1109/TIFS.2006.873653

29. Frederick Jelinek, Lalit R. Bahl, and Robert L. Mercer. 2006. Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech. IEEE Trans. Inf. Theor. 21, 3 (Sept. 2006), 250–256. DOI: http://dx.doi.org/10.1109/TIT.1975.1055384

30. Alisher Kholmatov and Berrin Yanikoglu. 2009. SUSIG: An On-line Signature Database, Associated Protocols and Benchmark Results. Pattern Anal. Appl. 12, 3 (Sept. 2009), 227–236. DOI: http://dx.doi.org/10.1007/s10044-008-0118-x

31. Hyeon-Kyu Lee and J. H. Kim. 1999. An HMM-based Threshold Model Approach for Gesture Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 10 (Oct 1999), 961–973. DOI: http://dx.doi.org/10.1109/34.799904

32. Yang Li. 2010. Protractor: A Fast and Accurate Gesture Recognizer. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). ACM, New York, NY, USA, 2169–2172. DOI: http://dx.doi.org/10.1145/1753326.1753654

33. Alexander De Luca and Janne Lindqvist. 2015. Is secure and usable smartphone authentication asking too much? Computer 48, 5 (May 2015), 64–68. DOI: http://dx.doi.org/10.1109/MC.2015.134

34. Michael D. Morse and Jignesh M. Patel. 2007. An Efficient and Accurate Method for Evaluating Time Series Similarity. In Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data (SIGMOD '07). ACM, New York, NY, USA, 569–580. DOI: http://dx.doi.org/10.1145/1247480.1247544

35. Miguel A. Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. 2013. Memorability of Pre-designed and User-defined Gesture Sets. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, 1099–1108. DOI: http://dx.doi.org/10.1145/2470654.2466142

36. Uran Oh and Leah Findlater. 2013. The Challenges and Potential of End-user Gesture Customization. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, 1129–1138. DOI: http://dx.doi.org/10.1145/2470654.2466145

37. Javier Ortega-Garcia, J Fierrez-Aguilar, D Simon, J Gonzalez, M Faundez-Zanuy, V Espinosa, A Satue, I Hernaez, J-J Igarza, C Vivaracho, and others. 2003. MCYT Baseline Corpus: A Bimodal Biometric Database. IEE Proceedings-Vision, Image and Signal Processing 150, 6 (2003), 395–401. DOI: http://dx.doi.org/10.1049/ip-vis:20031078

38. Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. 2000. Speaker Verification Using Adapted Gaussian Mixture Models. Digit. Signal Process. 10, 1 (Jan. 2000), 19–41. DOI: http://dx.doi.org/10.1006/dspr.1999.0361

39. Danuta Rutkowska. 2004. Multi-expert Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, 650–658. DOI: http://dx.doi.org/10.1007/978-3-540-24669-5_85

40. Napa Sae-Bae, Kowsar Ahmed, Katherine Isbister, and Nasir Memon. 2012. Biometric-rich Gestures: A Novel Approach to Authentication on Multi-touch Devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 977–986. DOI: http://dx.doi.org/10.1145/2207676.2208543

41. Alireza Sahami Shirazi, Peyman Moghadam, Hamed Ketabdar, and Albrecht Schmidt. 2012. Assessing the Vulnerability of Magnetic Gestural Authentication to Video-based Shoulder Surfing Attacks. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 2045–2048. DOI: http://dx.doi.org/10.1145/2207676.2208352

42. Florian Schaub, Marcel Walch, Bastian Könings, and Michael Weber. 2013. Exploring the Design Space of Graphical Passwords on Smartphones. In Proceedings of the Ninth Symposium on Usable Privacy and Security (SOUPS '13). ACM, New York, NY, USA, Article 11, 11:1–11:14 pages. DOI: http://dx.doi.org/10.1145/2501604.2501615

43. Muhammad Shahzad, Alex X. Liu, and Arjmand Samuel. 2013. Secure Unlocking of Mobile Touch Screen Devices by Simple Gestures: You Can See It but You Can Not Do It. In Proceedings of the 19th Annual International Conference on Mobile Computing & Networking (MobiCom '13). ACM, New York, NY, USA, 39–50. DOI: http://dx.doi.org/10.1145/2500423.2500434

44. Michael Sherman, Gradeigh Clark, Yulong Yang, Shridatt Sugrim, Arttu Modig, Janne Lindqvist, Antti Oulasvirta, and Teemu Roos. 2014. User-generated Free-form Gestures for Authentication: Security and Memorability. In Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '14). ACM, New York, NY, USA, 176–189. DOI: http://dx.doi.org/10.1145/2594368.2594375

45. Jing Tian, Chengzhang Qu, Wenyuan Xu, and Song Wang. 2013. KinWrite: Handwriting-Based Authentication Using Kinect. In Proc. of NDSS '13.

46. Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. 2013. Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS '13). ACM, New York, NY, USA, 161–172. DOI: http://dx.doi.org/10.1145/2508859.2516700

47. Umut Uludag, Sharath Pankanti, Salil Prabhakar, and Anil K. Jain. 2004. Biometric cryptosystems: issues and challenges. *Proc. IEEE* 92, 6 (June 2004), 948–960. DOI: http://dx.doi.org/10.1109/JPROC.2004.827372

48. Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2012. Gestures As Point Clouds: A $P Recognizer for User Interface Prototypes. In Proceedings of the 14th ACM International Conference on Multimodal Interaction (ICMI '12). ACM, New York, NY, USA, 273–280. DOI: http://dx.doi.org/10.1145/2388676.2388732

49. Radu-Daniel Vatavu, Daniel Vogel, Géry Casiez, and Laurent Grisoni. 2011. Estimating the Perceived Difficulty of Pen Gestures. In Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II (INTERACT'11). Springer Berlin Heidelberg, Berlin, Heidelberg, 89–106. DOI: http://dx.doi.org/10.1007/978-3-642-23771-3_9

50. Emanuel von Zezschwitz, Paul Dunphy, and Alexander De Luca. 2013. Patterns in the Wild: A Field Study of the Usability of Pattern and Pin-based Authentication on Mobile Devices. In Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '13). ACM, New York, NY, USA, 261–270. DOI: http://dx.doi.org/10.1145/2493190.2493231

51. Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures Without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. In Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07). ACM, New York, NY, USA, 159–168. DOI: http://dx.doi.org/10.1145/1294211.1294238

52. Junshuang Yang, Yanyan Li, and Mengjun Xie. 2015. MotionAuth: Motion-based authentication for wrist worn smart devices. In Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on. 550–555. DOI: http://dx.doi.org/10.1109/PERCOMW.2015.7134097

53. Yulong Yang, Gradeigh D. Clark, Janne Lindqvist, and Antti Oulasvirta. 2016. Free-Form Gesture Authentication in the Wild. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '16). ACM, New York, NY, USA, 3722–3735. DOI: http://dx.doi.org/10.1145/2858036.2858270

54. Ziming Zhao, Gail-Joon Ahn, Jeong-Jin Seo, and Hongxin Hu. 2013. On the Security of Picture Gesture Authentication. In Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13). USENIX, Washington, D.C., 383–398. https://www.usenix.org/conference/usenixsecurity13/technical-sessions/presentation/zhao

55. Nan Zheng, Kun Bai, Hai Huang, and Haining Wang. 2014. You Are How You Touch: User Verification on Smartphones via Tapping Behaviors. In 2014 IEEE 22nd International Conference on Network Protocols. IEEE, 221–232. DOI: http://dx.doi.org/10.1109/ICNP.2014.43