To appear in the Proceedings of IEEE BTAS, September 29-October 2, 2013, Washington DC, USA

Which Verifiers Work?: A Benchmark Evaluation of Touch-based Authentication Algorithms

Abdul Serwadda, Vir V. Phoha, Zibo Wang
Louisiana Tech University, Ruston, LA 71272
{ase007,phoha,zwa006}@latech.edu

Abstract

Despite the tremendous need for the evaluation of touch-based authentication as an extra security layer for mobile devices, the huge disparity in the experimental methodology used by different researchers makes it hard to determine how much research in this area has progressed. Critical variables such as the types of features and how they are pre-processed, the training and testing methodology and the performance evaluation metrics, to mention but a few, vary from one study to the next. Additionally, most datasets used for these evaluations are not openly accessible, making it impossible for researchers to carry out comparative analysis on the same data.

This paper takes the first steps towards bridging this gap. We evaluate the performance of ten state-of-the-art touch-based authentication classification algorithms under a common experimental protocol, and present the associated benchmark dataset for the community to use. Using a series of statistical tests, we rigorously compare the performance of the algorithms, and also evaluate how the "failure to enroll" phenomena would impact overall system performance if users exceeding certain EERs were barred from using the system. Our results and benchmark dataset open the door to future research that will enable the community to better understand the potential of touch gestures as a biometric authentication modality.

1. Introduction

Over the past few years, the popularity and usage of mobile devices (*i.e*. smart phones, tablets, *etc*.) has grown exponentially [2]. One of the key factors for the proliferation of these devices—their portability relative to the desktop computer—also unfortunately manifests as a major weakness from the point of view of physical security. The ease with which these devices can be carried around in their owners' pockets and (or) briefcases is the same ease with which they can be misplaced or stolen by adversaries. Once in the hands of a sophisticated attacker, both the remotely accessible resources and stored data on these devices (*e.g*. passwords, social security numbers, bank details, private emails, company secrets, etc) could easily be compromised, potentially resulting into catastrophic consequences for businesses and (or) individuals.

Currently, the most widely employed defense against such threats is the PIN lock mechanism. However, this mechanism is incorrectly used by some users (*e.g*. by setting very long timeouts [10]), completely disengaged by others [16], and susceptible to several attacks even when users engage it in accordance with the best practices (*e.g*. see [4][15]). To augment the single line of defence offered by the PIN lock, researchers have recently explored the possibility of *continuously* authenticating users after the initial login phase is completed [11].

One continuous authentication approach that has attracted a lot of interest revolves around touch gestures that users execute during their routine operations on the phone (*e.g*. see [10][13][20]). Touch gestures arise naturally from operations such as scrolling, zooming and clicking, and can thus be used by an authentication application without requiring the user to pay attention to the authentication process.

In a recent Active Authentication (AA) research drive championed by the Defense Advanced Research Projects Agency (DARPA) (see [1]), touch gestures have been identified as one of the candidate biometric modalities that could be built into a pilot multi-modal "biometric platform [1]" to be deployed in IT devices at the Department of Defense (DoD). Clearly, the need to examine the potential of touch gesture-based continuous authentication is now a research problem that is of profound interest to not only academicians, but to government and industry as well.

While recent works have demonstrated some ground breaking results on the applicability of touch gestures for continuous authentication (*e.g*. see [10][13][20]), the huge disparity in the experimental conditions and performance evaluation methodology used by different researchers makes it very difficult to put the various published results into context. Several variables that are integral to the perceived per-

formance of a biometric system, such as: 1) the composition of feature vectors used to build users' templates, 2) the number of instances used for training, 3) the way in which data is preprocessed —*e.g*. how or whether outliers are filtered off, and, 4) the classification thresholds used to report results, to mention but a few, vary from one study to another. To compound this challenge even further, each study is based on a different dataset, making it very hard for researchers to carry out meaningful comparisons between their methods and (or) findings.

Regarding the question of which verification algorithms are most suited for continuous touch based authentication, there also still exists a research gap since all past works have been based on a very small set of classification algorithms (typically 1 to 3 algorithms per study— *e.g*. see [10][13][20]). This tendency for "touch" authentication researchers to focus on a very small set of classifiers has left a large number of well known high performance algorithms unexplored, both in their individual capacities, and in terms of how they compare amongst each other.

For the community to fully understand the true potential of "touch" biometrics, there is a dire need for a benchmark study that evaluates the performance of a wide range of classification algorithms under a common experimental protocol. To ensure that future research can easily build new innovations off of such a study, it is not only important that the study be based on a publicly accessible dataset, but its also crucial that thorough analysis be made, with the full experimentation procedure laid out in details for other researchers to follow or modify accordingly. *The previous two statements are the principal motivation for this study*. Specifically, the paper makes the following contributions:

1. Based on a large *publicly accessible* touch biometrics dataset [1], we evaluate the performance of *ten* state-of-the-art classification algorithms, five of which never studied before for touch based authentication. We compute the Equal Error Rates (EERs) of the algorithms, and use a series of statistical tests to compare the performance of the algorithms under a common experimental protocol.

2. For the three best-performing verification algorithms, we carry out a user-level analysis of the error rates and show results on how a *failure-to-enroll* policy implemented on users exceeding a certain EER threshold would impact overall system performance. At different EER thresholds, we show the number of users who manage to enroll and the new system EER based on the smaller subset of users. This kind of analysis, never done before for touch-based authentication, provides interesting insights into the proportion of users whose touch behavior is consistent enough to be suited for touch based authentication technologies.

The rest of the paper is organized as follows. We discuss related work in Section 2, give a detailed description of our experiments in Section 3, and present the results of our performance evaluation in Section 4. We finally make our conclusions in Section 5.

2. Related Work

Research on touch-based authentication can be categorized into two groups: 1) authentication mechanisms in which touch gestures are used for authentication at an entry point (*e.g*. at login), and, 2) authentication mechanisms in which touch gestures are extracted continuously as the user performs various tasks on the phone. The former category includes studies in which users touch behavior is analyzed based on a set of canonically defined gestures (*e.g*. see [17][18] or gestures strictly captured at the unlock screen (*e.g*. see [8]).

"Entry point" touch-based authentication has several operational dissimilarities with continuous touch-based authentication. Perhaps the most notable of these is the fact that the known geometry of the hand can be easily matched with the strictly defined structure of a gesture to ensure that only touch points associated with similar fingers (say, a thumb in the template and a thumb presented during testing) are compared during "entry point" authentication (*e.g*. see [17]). Such kinds of assumptions can not be made with continuous authentication where users freely interact with the phones, touching them with whatever fingers and in whatever way they find comfortable. Due to space limitations and the fact that comparisons across the two types of authentication may be meaningless for most purposes, we only discuss results from past works which studied continuous touch-based authentication.

Table 1 gives a highlight of the experimental conditions and results from three studies which to our knowledge are the only works which have previously evaluated *continuous* touch-based authentication. From the small selection of attributes captured in the table, the gap that this paper seeks to bridge is apparent—the number of features used varies from 10 to 53 across the three studies, the identity of features used is not disclosed in [20], while feature selection is applied in [10] and [13] but not in [20]. EERs are used to report the performance in [10] and [13][2], while in [20] the authors report error rates based on a threshold they set heuristically. Feature normalization is performed in [10], but not in [20] and [13].

[1]The data used in this work can be accessed at—http://www2.latech.edu/ phoha/BTAS-2013.htm

[2]Li et al. [13] did not explicitly report EERs. We inferred EER estimates from their FAR/FRR vs block-size plots. The tabulated EER estimates (see Table 1) are based on the FAR and FRR corresponding to a block size of 20.

Study	# of Users	# of Features Used	Are the Features Identified?	Is Feature Selection Used?	Are the Features Normalized?	Is Outlier Filtering Done?	Classifiers	FAR (%)	FRR(%)
Frank et al. [10]	41	27	Yes	Yes	Yes	Short strokes removed	SVM	0.0–4.00	0.0–4.00
							kNN	0.0–4.00	0.0–4.00
Li et al. [13]	75	10	Yes	Yes	No	No	SVM	≈3.00	≈3.00
Feng et al. [20]	40	53	No	No	No	No	J48 Tree	≈14.00	≈12.00
							Random Forest	≈7.50	≈8.00
							Bayes Net	11.96	8.53

Table 1: Summary of results and evaluation conditions from the three studies which have previously explored continuous touch-based authentication on mobile phones. In [20], two sets of experiments are conducted: one with users wearing a special digital sensor glove, and the other with users touching the screen with their bare hands. We only tabulate results obtained for the case when bare hands were used since it is the more realistic phone usage scenario. In cases where we quote an approximate FRR/FAR (*i.e*. using the symbol ≈), its because we estimated the value from a graph in the paper in question. In [10], EERs of between 0.0 - 4.0% are reported. For ease of comparison with other works, we tabulate this EER range as an FRR of 0.0% - 4.0% at an FAR of 0.0% - 4.0%. Frank et al. [10] defined 30 features and discarded three of them after feature analysis, while Li et al. [13] defined 13 features and discarded 3 of them after feature analysis.

If one were tasked to quote the *typical* EER range of a continuous touch-based authentication, it would be very hard to deliver a precise answer based on the three studies. Likewise, if one were tasked to determine which kinds of features give the best performance for touch authentication, the high variance in experimental conditions across the three studies would make it very hard to give an informed solution. While this paper does not provide definitive answers to these kinds of questions, we believe that our findings and benchmark dataset represent a crucial first step based on which researchers will eventually provide more precise answers to these kinds of questions.

3. Experimental Design

3.1. Data Used for Experiments

For data collection we developed two Android applications through which users answered a series of multiple choice questions. To be able to answer the questions, users had to scroll/swipe back and forth to find the (short) paragraphs of text and (or) images on which questions were based.

For each of a set of points on a stroke representing the path taken by a user's finger during scrolling, the applications recorded: 1) the x and y coordinates, 2) the pressure exerted on the screen, 3) the area occluded between the finger and the screen, and, 4) the time at which the finger touched the point in question. We only captured these five raw features from two types of finger strokes: strokes for which the finger tip slides vertically (or approximately vertically) over the screen and strokes for which the finger tip slides horizontally (or approximately horizontally) over the screen. During typical phone usage, these strokes occur frequently as users browse pages to read text, or switch between two screens (*e.g*. while viewing images). This makes these kinds of strokes more suitable for a continuous authentication application than those associated with operations such as zooming [10]. We classified a stroke as horizontal if its horizontal displacement exceeded its vertical displacement. If the vertical displacement exceeded the horizontal displacement we classified the stroke as vertical.

We collected "touch" data from 190 unique subjects, all of whom students, faculty or staff at our university. Every user provided data over two sessions that were at least 1 day apart, with each session based on a different set of questions (*i.e*. a different application). We will respectively refer to the first and second data collection sessions as *Session I* and *Session II* throughout the rest of the paper. To avoid any bias that might have arisen from differences in variables such as the screen sizes and resolutions across different phone types, we used a single model of phone (*Google Nexus S* running Android version 4.0) for all data collection.

3.2. Classification Process

3.2.1 Computing Feature Vectors

We flagged all short strokes (having four or less touch points) as outliers and discarded them before proceeding to the feature vector computation step. A similar stroke filtering step was undertaken in [10]. Following outlier removal, we computed several statistical measures that we used to build a 28-dimensional feature vector representing each stroke. Below, we briefly describe how each of the 28 features were computed.

For every pair of adjacent points on a stroke, we computed the velocity along the stroke. Two adjacent velocity values were then used to compute an acceleration value. From the vector of velocity values computed for a stroke, we computed the mean value, standard deviation, first quar-

tile, second quartile and third quartiles — five measures that provide insights into the distribution of values in each of the vectors. For each of the acceleration, pressure and area measurements associated with a stroke, we computed the same five measures. The values computed from this process represent 20 (=5×4) of the 28 features representing a stroke. The final 8 features completing the vector were: the coordinates of the extreme points of a stroke (this gives a total of four features since each x,y pair is considered as two features)[3], the distance between the start and end points of a stroke, the total time taken to complete a stroke, the tangent of the angle between the line joining the end-points of a stroke and the horizontal, and the summation of the distances between every pair of adjacent points along the stroke.

We processed the portrait strokes separately from the landscape strokes (i.e., strokes respectively executed when the phone was held in portrait and landscape orientation) because certain features — e.g., coordinates of start and end points, lengths of strokes, etc. — may change with changes in phone orientation even when the user in question is the same. For users who had both portrait and landscape strokes, we thus built separate reference templates for either family of strokes. For each of the portrait and landscape orientations, we further distinguished between vertical and horizontal strokes and built separate templates for each.

This way, we ensured that a stroke presented for authentication would be compared with the corresponding kind of template for the user in question. In practice, we believe that a continuous touch authentication application should build all four kinds of reference templates for each user because for each of the vertical and horizontal families of strokes, one could easily switch between the two orientations depending on the design dynamics (e.g., fonts, organization of pages, etc.) of the content that the user reads on the phone at a particular point in time.

3.2.2 Training and Testing Details

For each participant, we used data collected during *Session I* for training, and data collected during *Session II* for testing. For classifiers requiring only instances of the positive class for training, we used 80 strokes (executed by the target user) during the training phase. For classifiers requiring instances of both the positive and negative classes for training, we used 80 strokes from the target user and 80 strokes randomly chosen impostors. Users who did not execute at least 80 strokes for a certain category of strokes (e.g., the category of horizontal strokes under portrait orientation) during *Session I* were excluded from our performance evaluation for the category of strokes in question. We enforced the 80 strokes requirement so as to avoid any performance bias that might have arisen out of certain users having much larger (or smaller) numbers of training vectors in comparison to others. Table 2 summarizes the number of users who met our 80 strokes requirement for each family of strokes.

	Portrait		**Landscape**	
	Horizontal	Vertical	Horizontal	Vertical
Number of Users	106	118	41	50

Table 2: Number of users who executed at least 80 strokes for each category of strokes.

Genuine (or positive class) testing was based on all strokes executed by the target user during *Session II*. For impostor (or negative class) testing, impostor samples were drawn in such a way that each user (besides the user whose model was being tested) contributed ten instances to the impostor set used to attack a given user's template.

During testing, we used a sliding window mechanism in which *n* vectors (*i.e*. feature vectors derived from *n* distinct strokes) were used to compute a single feature vector that was used for testing. Each vector in the test set was hence derived from *n* distinct strokes. The single vector was such that the element having index i was computed as the mean of the i^{th} elements of the *n* vectors. Our observation was that authentication attempts made based on a block (window) of strokes gave a more coherent biometric footprint, and hence better performance than those based on individual strokes. For all classifiers we used *n*=10. Features were subjected to min-max normalization (to the interval 0-1).

3.2.3 Classifiers Under Investigation

We compared ten classification algorithms, eight of which widely used in general machine learning research, and two, mostly studied in keystroke and mouse dynamics. Our motivation for studying the latter (two) algorithms was due to the conjecture that a set of classification algorithms that have previously been shown to handle the high intra-user variability and low inter-user variability seen with mouse and keystroke dynamics could perform fairly well if applied to the similarly behaved touch gestures. It was thus interesting to see how these algorithms compared with the more main stream machine learning algorithms.

The eight well known algorithms that we used are: Support Vector Machines (SVM) [6], Naïve Bayes [9], Random Forests [5], k-Nearest Neighbors (kNN) [7], Bayesian Networks [9], Neural Network [9] (*i.e*. Multi-layer Perceptron), J48 tree [9] and Logistic Regression used as a classifier [21]. Five of these algorithms — SVM, kNN, J48, Random

[3]For the horizontal strokes, these coordinates are the extreme right and extreme left most points. For the vertical strokes, these coordinates are the extreme top and extreme bottom points of a stroke.

Forests and Bayesian Networks—have previously been studied in various works in the touch-based continuous authentication realm (see [20], [13], [10]). This work hence benchmarks their performance on a much larger dataset, under a set of common experimental conditions. Due to space limitations, we do not delve into the mechanisms of operation of these eight algorithms since they are very widely studied. We however briefly discuss the two mouse and keystroke dynamics algorithms which, despite building around well known pattern recognition principles, have for the most part been used by a relatively small section of the biometrics community. A brief description of the training and testing routines undertaken by these algorithms follows:

Scaled Manhattan Verifier [12][19]: During training, this algorithm computes the mean and mean absolute deviation of each feature. In the test phase, the score of a given test vector is computed using the expression $\Sigma_{i=1}^{p}|a_i - b_i|/y_i$, where a_i is the value of the i^{th} feature in the test vector, b_i is the value of the i^{th} feature in the mean vector computed during training, and y_i is the absolute average deviation obtained during training.

Euclidean Verifier [12][19]: During training the classifier computes the mean value of each feature. During testing, the score for a given test vector is the squared Euclidean distance between the test vector and the mean vector computed during training.

Classifier	Portrait		Landscape	
	Horizontal	Vertical	Horizontal	Vertical
Logistic Regression	13.8(11.6)	17.2(14.0)	10.5(9.2)	13.7(11.2)
SVM	15.7(13.6)	18.0(13.6)	13.1(13.1)	14.7(9.8)
Random Forest	14.7(13.4)	21.7(16.5)	12.7(8.7)	16.7(13.6)
Naïve Bayes	17.8(16.7)	25.0(17.3)	20.4(19.6)	18.8(14.3)
Multilayer Perceptron	16.0(16.9)	20.7(16.1)	14.8(15.4)	14.8(12.1)
kNN	17.3(14.4)	27.1(18.1)	14.0(9.8)	20.1(15.2)
Bayesian Network	18.2(16.6)	26.0(18.4)	16.2(10.8)	17.4(13.0)
Scaled Manhattan	21.9(15.7)	28.7(14.7)	23.1(15.3)	22.2(13.8)
Euclidean	24.1(17.2)	19.5(19.1)	23.5(15.2)	23.7(13.1)
J48	30.7(28.6)	38.0(28.4)	42.0(33.8)	33.2(13.8)

Table 3: Mean EER and standard deviation of the EERs across the population for each of the ten classifiers. The EERs have been expressed as percentages (*i.e.* on a scale running from 0 to 100). For each of the portrait and landscape orientations, results from the classification of the vertical and horizontal strokes are tabulated separately.

4. Performance Evaluation

4.1. General Performance Across Population

To analyze the performance of the verification algorithms, we use the Equal Error Rate (EER), a measure that is widely used for biometric performance evaluation (*e.g.* see [13][10]). The EER is the error rate at which the probability of false acceptance is equal to the probability of false rejection. The lower the EER is the better the performance of a biometric system. For each classification algorithm, we computed the EER for each user, and then calculated the mean and standard deviation of EERs over the population. Table 3 summarizes these results for all ten classifiers. The SVM verifier is based on the Gaussian Radial Basis function, while the Random Forest is built using 1000 trees. The k-NN verifier is set with k=9. For the eight well known algorithms, all other settings are the default settings in WEKA [21]. For the keystroke and mouse dynamics verifiers, no other settings are required on top of the mechanisms described in Section 3.2.3.

Observe (Table 3) that the Logistic Regression classifier and the J48 tree respectively had the lowest and highest mean EERs across all stroke categories. Overall, our mean EERs are comparable to those reported in [20], but quite distinct from those obtained in [10] and [13]. There is a large array of factors that could be responsible for this disparity in performance. These include differences in the feature definitions, feature pre-processing methods, number of instances used for training and the compositions and sizes of the study populations to mention but a few. As mentioned earlier, we believe that our results and benchmark dataset will make it possible for researchers to analyze the impact of these factors. This should in turn enable the community to get a unified view of how continuous touch based authentication could perform at its very best.

Note that while the means and standard deviations of the EERs reported in Table 3 give an interesting summary of how the classifiers performed, a more insightful view of how the classifiers compared amongst each other has to be based on analysis of the full distribution of EERs. Details and findings from this analysis are given in the next section.

4.2. Ranking the Algorithms

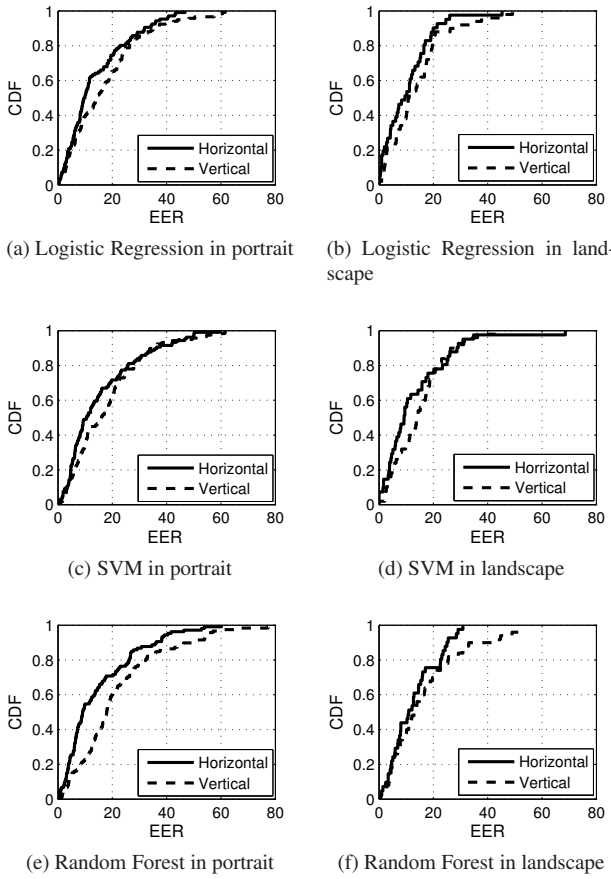
Let EER_{svm} denote the vector of the EERs associated with the SVM verifier. Each element in EER_{svm} is the EER obtained for a distinct user in our population when the SVM was used for classification. Let EER_{lr} denote a similar vector for the Logistic Regression classifier. To determine whether the Logistic Regression classifier performed better than the SVM, we are interested in determining if the elements of vector EER_{svm} are significantly greater than those in vector EER_{lr}. For this purpose a paired t-test performs well if the vector of paired differences (EER_{svm}-

EER_{lr}) exhibits Gaussian behavior. Past work used either graphical means (e.g., see [12]) or statistical significance testing (e.g., see [3]) to check for conformity to the normality assumption. In this work we used the latter approach. Our full ranking method, based on the method used in [12], is next described in details.

We designated the verification algorithm having the lowest mean EER (*i.e*. Logistic Regression) as the best performing algorithm, and then compared each of the other nine algorithms with it. For each of the nine algorithm pairings (for each family of strokes) we first subjected the associated vector of EER differences to the Kolmogorov Smirnov (K-S) normality test [14] under the null hypothesis of the differences vector being Gaussian. At the 5% significance level, we rejected the null hypothesis in all tests performed. With the EER vectors being non-Gaussian, we decided to use the Wilcoxon signed rank test to compare the performance of the verification algorithms. In each test, the null hypothesis was that the best performing classifier was not any better than the algorithm with which it was being compared.

When we run the tests on the EER vectors that were obtained for the horizontal strokes (for each of the portrait and landscape orientations), we failed to reject the null hypothesis at the 5% significance level for the Random Forest and SVM. For all other algorithms, we rejected the null hypothesis. For the horizontal strokes therefore, the SVM, Random Forests and Logistic Regression algorithms were the best performers in our study.

When we run the same tests on the EER vectors that were obtained for the vertical strokes, we rejected the null hypothesis in all cases. In this case therefore, no verification algorithm was comparable to the Logistic Regression algorithm.



(a) Logistic Regression in portrait (b) Logistic Regression in landscape

(c) SVM in portrait (d) SVM in landscape

(e) Random Forest in portrait (f) Random Forest in landscape

Figure 1: CDF of the EERs obtained across the full population.

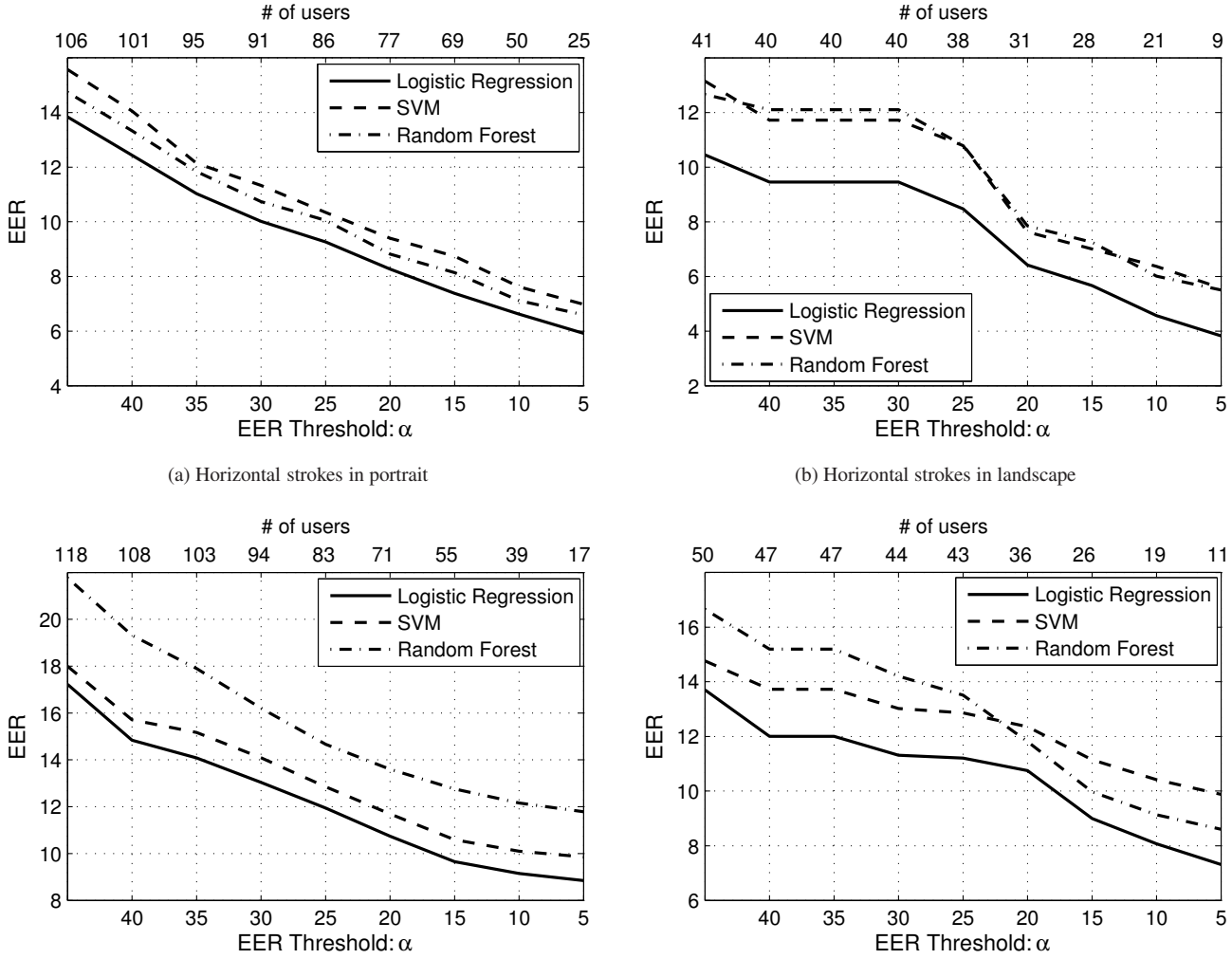
4.3. User-level Performance Evaluation

Although the "*failure to enroll"* concept is widely used in physical biometrics, security evaluations of behavioral biometric modalities rarely put this concept into consideration. Given the well known instability of behavioral biometric traits for certain categories of users, its natural to ask the questions: If users whose mean performance is worse than a certain threshold were to be barred from using the system, 1) how much would the overall system performance improve ?, 2) what proportion of the population would still be able to use the technology?

These kinds of questions prompted us to make a user-level analysis of the performance, and to evaluate how systematic exclusion of the "bad" users would affect system performance. Due to space limitations, we only perform this analysis based on the three verifiers which had the lowest mean EERs. Figure 1 shows the CDFs of the EERs obtained with the three verifiers. In all 6 plots, it is apparent that there is a small group of users who had really high EERs (e.g., about 20% to 30% of the participants had EERs exceeding 20%). This trait is seen across both phone orientations.

Figure 2 shows how systematic removal of the "bad" users affects system performance. For each EER threshold α, we remove all users who had a mean EER greater than α across all three classifiers. After the bad users are removed, we recompute the new mean EER for each algorithm. The value of the new EER after each step is shown on the X-axis at the bottom of the page, while the number of users left after the poor users are removed at each stage is shown on the axis at the top of the plot. We used values of α ranging from 40% to 5%.

Observe that at $\alpha = 20\%$, the mean EER for the horizontal strokes is under 10% for all verifiers, while more than 70% of the initial population (i.e., 77 out of 106 users for portrait and 31 out of 41 users for the landscape strokes) are able to enroll onto the system. At $\alpha = 10\%$ the number of users enrolled on the system has drastically dropped



(a) Horizontal strokes in portrait

(b) Horizontal strokes in landscape

(c) Vertical strokes in portrait

(d) Vertical strokes in landscape

Figure 2: How systematic removal of poor users affects the mean EER of each verifier. At a given EER threshold α, all users whose mean EER (computed based on the individual EERs obtained for the four verifiers) exceeds α are removed and the new mean EER for each algorithm computed again. The bottom axis shows the thresholds, α, while the top axis shows the number of users left after a group of users is removed based on the threshold α. The vertical axis shows the new EER after the poor users are removed.

to about 50% of the initial population. However, the mean EER has also dropped to under 7% for the landscape (horizontal) strokes and to under 8% for the portrait (horizontal) strokes. For each value of α, the performance gains due to the "*failure to enroll"* policy are more pronounced with the horizontal strokes, however, the overall benefits of the policy are still apparent. While the question of the optimal threshold EER is a research problem beyond the scope of this paper, these results provide the first indication of the cost-benefit trade-off associated with a "failure to enroll" policy in continuous touch-based authentication.

5. Conclusions

In this paper we have performed a benchmark evaluation of ten classification algorithms for touch based authentication. We have laid out a repeatable evaluation procedure, and presented a large touch biometrics dataset for the community to use. While all ten verification algorithms had mean EERs of over 10% during our initial evaluation, we found that a "failure to enroll" policy that excludes users whose EERs exceeded certain thresholds improved performance tremendously. For future research, we plan to explore how selective fusion strategies that use multiple bio-

metric modalities for the sub-set of poor users affect overall performance.

6. Acknowledgement

This work was supported in part by the Louisiana Board of Regents under P-KSFI Grant LEQSF(2007-12)-ENHPKSFI-PRS-03.

References

[1] Darpa-baa-13-16 active authentication (aa) phase 2. https://www.fbo.gov/index?s=opportunity&mode=form&id=aa99ff477192956bd706165bda4ff7c4&tab=core&_cview=1. Last accessed in April, 2013.

[2] Gartner says worldwide pc, tablet and mobile phone combined shipments to reach 2.4 billion units in 2013. http://www.gartner.com/newsroom/id/2408515. Last accessed in April, 2013.

[3] Multitask learning for hostpathogen protein interactions. *Bioinformatics*, 29(13):i217–i226, jul 2013.

[4] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. In *Proceedings of the 4th USENIX conference on Offensive technologies*, WOOT'10, pages 1–7, Berkeley, CA, USA, 2010. USENIX Association.

[5] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, Sept. 1995.

[7] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1):21–27, Sept. 2006.

[8] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 987–996, New York, NY, USA, 2012. ACM.

[9] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, 2nd edition, 2002.

[10] 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. pages 136–148, 2013.

[11] M. Jakobsson, E. Shi, P. Golle, and R. Chow. Implicit authentication for mobile devices. In *Proceedings of the 4th USENIX conference on Hot topics in security*, HotSec'09, pages 9–9, Berkeley, CA, USA, 2009. USENIX Association.

[12] K. S. Killourhy and R. A. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *DSN*, pages 125–134, 2009.

[13] L. Li, X. Zhao, and G. Xue. Unobservable reauthentication for smart phones. In *Proceedings of the 20th Network and Distributed System Security Symposium*, NDSS'13, Reston, VA, 2013. Internet Society.

[14] F. J. Massey. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

[15] E. Owusu, J. Han, S. Das, A. Perrig, and J. Zhang. Accessory: password inference using accelerometers on smartphones. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, HotMobile '12, pages 9:1–9:6, New York, NY, USA, 2012. ACM.

[16] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos. Progressive authentication: deciding when to authenticate on mobile phones. In *Proceedings of the 21st USENIX conference on Security symposium*, Security'12, pages 15–15, Berkeley, CA, USA, 2012. USENIX Association.

[17] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon. Biometric-rich gestures: a novel approach to authentication on multi-touch devices. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, CHI '12, pages 977–986, New York, NY, USA, 2012. ACM.

[18] N. Sae-Bae, N. Memon, and K. Isbister. Investigating multi-touch gestures as a novel biometric modality. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 156–161, 2012.

[19] C. Shen, Z. Cai, R. Maxion, G. Xiang, and X. Guan. Comparing classification algorithm for mouse dynamics based user identification. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 61–66, 2012.

[20] F. Tao, L. Ziyi, C. Bogdan, B. Daining, and S. Weidong. Continuous mobile authentication using touchscreen gestures. In *Proceedings of the 12th IEEE Conference on Technologies for Homeland Security*, HST'12, 2012.

[21] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.