Fast Continuous User Authentication using Distance Metric Fusion of Free-text Keystroke Data

Blaine Ayotte, Jiaju Huang, Mahesh K. Banavar, Daqing Hou, and Stephanie Schuckers
Department of Electrical and Computer Engineering, Clarkson University *
8 Clarkson Ave, Potsdam, NY 13699
{ayottebj, jiajhua, mbanavar, dhou, sschucke} @clarkson.edu

Abstract

Keystroke dynamics are a powerful behavioral biometric capable of determining user identity and for continuous authentication. It is an unobtrusive method that can complement an existing security system such as a password scheme and provides continuous user authentication. Existing methods record all keystrokes and use n-graphs that measure the timing between consecutive keystrokes to distinguish between users. Current state-of-the-art algorithms report EER's of 7.5% or higher with 1000 characters. With 1000 characters it takes a longer time to detect an imposter and significant damage could be done.

In this paper, we investigate how quickly a user is authenticated or how many digraphs are required to accurately detect an imposter in an uncontrolled free-text environment. We present and evaluate the effectiveness of three distance metrics individually and fused with each other. We show that with just 100 digraphs, about the length of a single sentence, we achieve an EER of 35.3%. At 200 digraphs the EER drops to 15.3%. With more digraphs, the performance continues to steadily improve. With 1000 digraphs the EER drops to 3.6% which is an improvement over the state-of-the-art.

1. Introduction

With the increase of sensitive and private data being stored online and on computers, protecting data has never been more important. Many devices requiring only a password or other form of one-time authentication can be breached or hacked by exploiting knowledge-based authentication [4]. Another form of authentication is needed to confirm that the user at the device is really the authorized user, on an ongoing basis.

Typed Text	Characters
Average tweet length	60-70
Average sentence	75-100
Phishing email	120
Average Facebook post	155
Maximum tweet length	280
Gettysburg Address	1450
Nigerian prince emails	1500-2500

Table 1. Estimates of character counts in different typed texts [9, 16, 22]. The dashed line separates texts with fewer than 1000 characters. Classification algorithms that require 1000 characters for intruder detection may miss attacks.

Keystroke dynamics are a behavioral biometric method offering strong performance for continuous user authentication [24, 25]. It is important to note that keystroke dynamics should not replace traditional authentication schemes, but rather complement existing ones. Keystroke dynamics are an extra layer of security that continuously authenticate users. If a user is logged into a device and somebody else starts using the device, it may be possible to detect an intruder and lock them out before serious damage can be done. This continuous authentication layer, not only provides extra security, but is also unobtrusive. Another strength of this additional security layer is that users need not change their routine behavior for the system to function. Furthermore, most computers already have a keyboard, so there is no requirement for additional hardware.

Many keystroke algorithms have been proposed using various techniques including hidden Markov models (HMM) [1], support vector machines (SVM) [26], and kernel density estimation (KDE) [7, 12]. Most algorithms are tested on a somewhat controlled dataset where users are given guidance on what to type. This guidance can range from giving users exact passages to type, to providing spe-

*Banavar and Ayotte are supported in part by the NSF CPS award 1646542. Ayotte is supported in part by the Clarkson Niklas Ignite Fellowship. This material is based upon work supported by the Center for Identification Technology Research (CITeR) and the National Science Foundation under Grants 1650503 and 1314792.

cific questions to answer. Less work has been done on datasets that are completely uncontrolled with no restrictions on user activity or typing behavior [12]. This scenario is of interest to us because it most closely resembles typical user behavior. In an uncontrolled environment, users could be switching between writing an essay, answering email, coding, playing video games, and many other activities without providing any explicit cues. Huang et al. investigated effects on the performance of free-text systems after removing what they called "gibberish" [13]. They noted that a possible explanation for some of the gibberish could be video games or other non-traditional text input activities. However, this type of behavior may be key to characterizing the user, making gibberish filtering subjective and likely to be different for each user as well as being hard to implement in a real-time system. For this paper no "gibberish" filtering is done to ensure all keystrokes are truly representative of user behavior.

In uncontrolled free-text environments, as expected, most algorithms perform worse overall [12]. For example, while multiple algorithms for keystroke dynamics using features only from passwords have high degrees of success (EER's under 1% with $\leq$ 30 fixed characters) [24], algorithms for strong performance on uncontrolled datasets need 500, 1000 or even more characters [2, 10, 11]. On average, one thousand characters is 10 sentences, 4 maximum length tweets, or a lengthy email that can be used for phishing (See Table 1). To enable faster and more frequent authentication, it is desirable that intruder detection be performed with much fewer characters, for example, on the order of a hundred. A system requiring 100 characters can detect intruders 10 times faster and thus 10 times more often than a system using 1000 characters. This enables continuous authentication systems to detect imposters far faster, better protecting user data.

In this paper, to investigate performance of keystroke dynamics with fewer keystrokes, we propose a modification on three existing digraph based algorithms. The first algorithm uses kernel density estimation (KDE) to derive a distance score between the reference and test empirical probability density functions (PDFs) [12]. This algorithm has achieved a great deal of success for long free-text, 1000 digraphs in the testing sample, and is comparable to other state-of-the-art algorithms [12]. The second algorithm is based on the Kolmogorov-Smirnov (KS) test and compares the cumulative distribution functions (CDFs) between the digraphs for the reference users and test users producing a similarity score. This metric has been previously used for larger test samples ($\geq$650 digraphs), but not yet for smaller test samples ($\leq$500 digraphs) [18]. The last algorithm uses an energy metric to compute a difference score between the reference and test PDFs. We perform fusion with different combinations of these distance metrics to evaluate our keystroke dynamics algorithms, individually and fused.

The rest of this paper is organized as follows. Section 2 describes the three metrics used. A brief summary of the uncontrolled free-text database used in this study and our results are provided in Section 3. The algorithms are evaluated using ROC curves, each computed with different amounts of digraphs in the testing sample. Fewer digraphs in the testing sample means the algorithm is faster. Finally, concluding remarks are presented in Section 4.

2. Algorithms

In this section, we focus on three distance metrics that all rely on the distributions of digraphs, *i.e.*, the frequency of occurrences of a given digraph versus flight time. Digraphs are the flight time between consecutive key-down presses and are commonly used in keystroke dynamics [24]. In contrast to the study in [21], we do not consider all digraphs as a single feature but instead treat them independently.

These three algorithms rely on statistical approaches that exploit similarities or differences in the empirical probability functions (PDF's) or the empirical cumulative distribution functions (CDF's). The PDF's and CDF's are generated from the distributions of digraphs, occurrence versus flight time. The CDF's are created as follows:

$$CDF(x) = \frac{1}{N}\sum_{i=1}^{N} I_{x_i<x}, \tag{1}$$

where N is the number of samples used to recreate the distribution, x is the flight time of the digraph, and I_A is the indicator function on event A [17]. The CDF's are created for the reference user from their training data and for the test user from a sample of testing data. The algorithms only compute CDF's when there at least four of the same digraph present in the sample, which provides a reasonable estimate of the CDF under ideal circumstances [12].

2.1. Kernel Density Estimation

The kernel density estimation (KDE) algorithm used in this paper is a modified version of the algorithm proposed in [12]. KDE is a non-parametric method used to estimate the PDF of a random variable. Here, it is used to create a PDF of the flight-times for each digraph from a finite number (> 4) of samples [20]. If there are less than four occurrences of a digraph in either the training sample or testing sample, that graph is not used. Given $N > 4$ samples of each digraph, the PDF is estimated at a point y within a group of points $x_i; i = 1, \ldots, N$ as

$$P_k(y) = \sum_{i=1}^{N} K\left(\frac{y - x_i}{h}\right), \tag{2}$$

where $K(x;h)$ is a kernel function controlled by the bandwith parameter h. We are using Gaussian kernel functions where $K(x;h) \propto \exp\left(-x^2/2h\right)$. P_k is estimated for both the training and testing data from the N digraph samples present. We used the python library scikit-learn's implementation of Gaussian KDE for PDF estimation [19]. Once the PDFs are estimated, the absolute difference of the PDFs for training and testing samples is calculated, summed, and then averaged across all the different digraphs to produce one scalar value. This averaged scalar value is a dissimilarity score between the training and testing samples. If the score is above a certain threshold, authentication fails.

2.2. Kolmogorov-Smirnov

The Kolmogorov-Smirnov (KS) test is a non-parametric statistical test used for comparing two independent or non-related samples [6]. The KS test computes the maximum absolute distance between two empirical cumulative distribution functions (CDFs) to give the quantity

$$K = \max_x |CDF_{train}(x) - CDF_{test}(x)|. \quad (3)$$

The empirical CDF is computed from the samples in the training and testing samples. A p-value is generated from the distance score using K and the data in the training and testing samples as

$$p = s\left(K\sqrt{\frac{n_1 n_2}{n_1 + n_2}}\right), \quad (4)$$

where $s(u) = \exp\left(-2u^2\right)$ [14], and n_1 and n_2 are the number of data points in the training and testing samples, respectively.

For a graph with only a few samples in the testing profile, the probability returned from the KS test is high regardless of whether the user is an imposter or not. To avoid this problem, only digraphs with four or more samples in both the training and testing data are used.

The scipy Python library is used to perform the KS test and generate the p-values with the `stats.ks_2samp()` function [8]. The p-values from each shared graph between training and testing samples are averaged together. This averaged p-value serves as a similarity score between the training and testing samples. If the score is below a certain threshold, the user is not authenticated and must authenticate themselves through other means.

2.3. Energy Distance

Energy statistics are functions of distances between statistical observations in metric spaces [23]. The energy distance is a non-parametric statistical test for comparing two distributions defined as

$$E = \left[\sum_x \{CDF_{train}(x) - CDF_{test}(x)\}^p\right]^{\frac{1}{p}} \quad (5)$$

When $p = 2$, the energy distance becomes the Cramer distance [3]. The energy distance is calculated for each graph that has four or more occurrences. The energy distance is a dissimilarity score and the closer the energy distance is to zero, the closer the two distributions are. The energy scores are averaged together for all shared graphs between training and testing samples. The test user is then either authenticated or deemed an imposter from the average energy score. The energy distance is computed using the python library, scipy, with the command `stats.energy_distance()` [8].

2.4. Distance Metric Fusion

To improve authentication accuracy at fewer keystrokes, we fuse the metrics from the three algorithms discussed previously. We compare different sets of fused metrics that use the KS, KDE and energy metrics one-at-a-time, in sets of two metrics, and all three metrics. This results in a total of 7 different classifiers. We consider two different fusion methods and report ROC curves for each. The first fusion method requires all three metrics to authenticate the test user for authentication. The second fusion method authenticates the test user when two or more of the three metrics authenticate the test user ("majority rules"). This fusion method is equivalent to a majority voting scenario. When only two metrics are fused the fusion decision requires both metrics to authenticate the user. Through fusion of the distance metrics we expect increased performance when the metrics contain independent information, i.e. the metrics are measuring different differences between users.

3. Evaluation and Results

In this section, we describe the dataset and the methods used to evaluate our algorithms, and compare our results to state-of-the-art algorithms. Results with testing samples ranging from 100 to 1000 digraphs are shown to demonstrate our algorithms fast performance and to best compare our algorithm to existing state-of-the-art algorithms. All methods of distance metric fusion are performed for the different testing sample sizes for a fair comparison.

3.1. Dataset

The data used for this study is the Clarkson II keystroke dataset collected through a study conducted at Clarkson University [15]. There are 103 users in the study and they contributed a combined 12.9 million keystrokes. To the best of our knowledge, this dataset is the largest available where an average user has 125k keystrokes. The keystrokes were recorded as long as the program was running regardless of application or context. Users had the option of temporarily disabling the keylogger to protect their private information.

Previous work on this dataset shows that the performance of algorithms can be improved by cleaning up "gibberish"

keystrokes from the data [13]. The authors coined the term "gibberish" to describe non-traditional typing behavior. A possible explanation for some of the "gibberish" is video game usage, coding, or other activities where keystrokes are not representative of traditional text entry practices such as writing an essay. Filtering out this "gibberish" might also result in losing some important user behavior or artificially even increasing performance. To ensure data is representative of a real-world setting, no "gibberish" filtering is done in this work.

3.2. Methods

One common approach to user authentication in keystroke dynamics is to use n-graphs and often only digraphs [24]. An n-graph is the timing between n consecutive keystrokes. In this work, we use $n = 2$, and a 2-graph is more commonly referred to as a digraph. Digraphs are calculated by looking at every pair of consecutive keystrokes. The flight times for the digraphs are calculated by subtracting the i^{th} and $i^{th} - 1$ keystrokes. Digraphs that occur over half a second apart are filtered out because it is likely the user has walked away from the keyboard or is not typing continuously. To ensure adequate data only users with at least 10,000 digraphs are used which leaves 79 users. The testing samples are randomly selected subsets of the 10,000 digraphs with 100, 200, 500, or 1000 consecutive digraphs. The training sample becomes 9000 of the remaining digraphs to ensure for each test sample size the same amount of training data is present. To better generalize results across our data, Monte Carlo analysis and cross validation is used with 20 random subsets of the data. Each subset contains different training and testing digraphs. This shows our results do not depend on particular subsets of the data and are in fact representative of the entire dataset.

The simulations were run on a computer with an Intel core i5 processor and took about one hour per Monte Carlo iteration. In each iteration, the model was retrained before testing, which led to the simulation taking considerable time. Each user was tested against themselves and all other users for every iteration. This results in many more imposter attacks, however, results are averaged when computing false accept and false reject rates so there is no adverse effect on error rates. With 79 users this equates to 79 genuine user attempts and 78 $\times$ 79 imposter attacks per Monte Carlo iteration. It is important to note that for every iteration, the training and testing data was different and all calculations were redone. Additionally, code was required to perform the random sampling of the dataset adding to total time of the simulation. The authentication algorithms by themselves are not that computationally expensive and could be implemented in real time for a single user with pre-collected training data.

Classification algorithms are used to detect if a user at a keyboard is the authorized user or an intruder/imposter. We use seven combinations of distance metric fusion for the classifiers as described in Section 2.4. The algorithms use data from the Clarkson II keystroke dataset. In what follows, the classification results are presented for the metrics individually and fused together.

3.3. Individual Metric Results

Figures 1, 2, and 3 show the ROC curves for the KDE, KS, and Energy algorithms, respectively. The KDE and Energy algorithms were the best performers, both with very similar performance. This result is consistent with previous

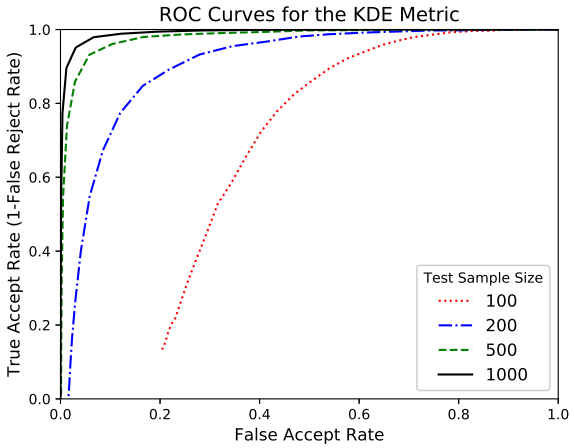

Figure 1. ROC curves for the KDE based algorithm with 100, 200, 500, and 1000 consecutive digraphs in the testing sample. As the testing sample size increases, the performance improves. For 100 digraphs in the testing sample, the EER is 35.8%.

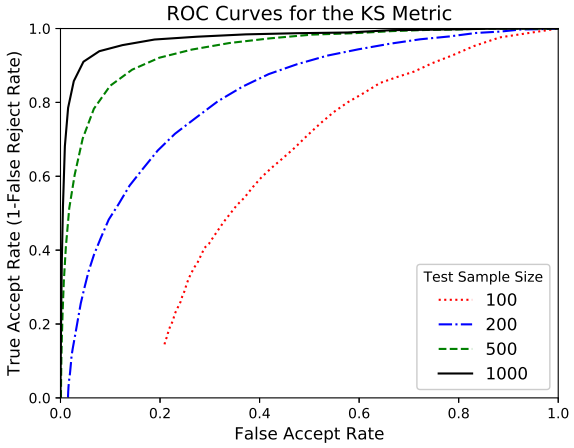

Figure 2. ROC curves for the KS based algorithm with 100, 200, 500, and 1000 consecutive digraphs in the testing sample. As the testing sample size increases, the performance improves. For 100 digraphs in the testing sample, the EER is 40.3%.

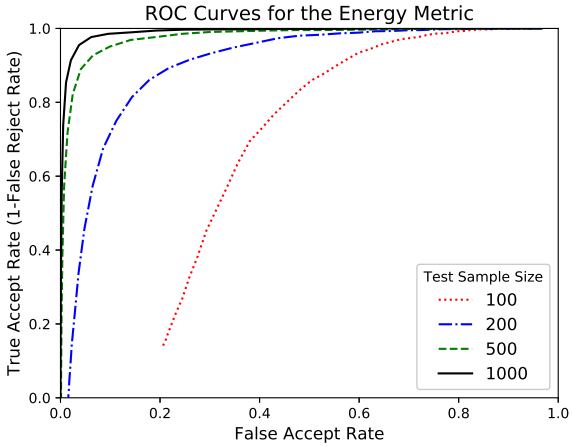

Figure 3. ROC curves for the Energy based algorithm with 100, 200, 500, and 1000 consecutive digraphs in the testing sample. As the testing sample size increases, the performance improves. For 100 digraphs in the testing sample, the EER is 36.1%.

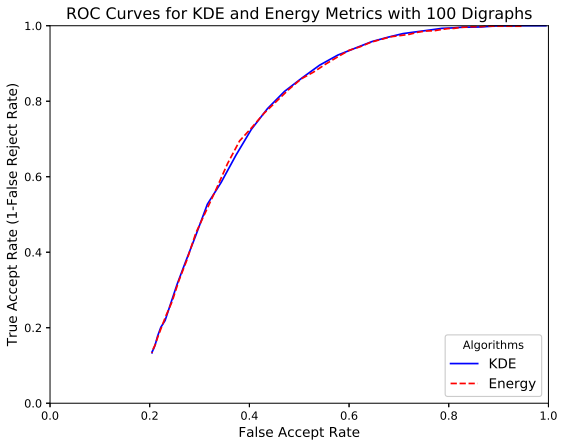

Figure 4. ROC curves for the KDE and Energy based algorithm with 100 consecutive digraphs in the testing sample. The two metrics are very similar with only slight deviation in few locations.

works where the KDE based algorithm performed strongly in uncontrolled free-text environments [12]. The KS algorithm has the worst overall individual performance although not much worse than KDE or Energy. The KDE and energy algorithms perform similarly since both metrics are calculated as a difference between an empirical PDF or empirical CDF of the same data. Figure 4 clearly shows the similarity between the KDE and Energy metrics. The two ROC curves are almost identical, only deviating slightly in a few locations. This is expected as the two metrics are computed very similarly, one from the absolute differences in training and testing PDF's and the other from the absolute differences in training and testing CDF's. Testing samples above 100 digraphs are even more similar and have less slight deviations. For the the rest of the paper, due to the similarity of the performance of the metrics, only the performance of the KDE algorithm is shown in the plots. We should note here that although the KDE and Energy metrics perform similarly, they are not identical. Therefore, the majority rules fusion rule (see Section 2.4) does not simply follow the performance of the KDE algorithm.

The ROC curves for smaller amounts of testing data (100 digraphs in testing sample) do not start at the origin, but at approximately 15% true accept rate and 21% false accept rate. This is not a problem for larger amounts of digraphs in the testing sample and the ROC curves start at the origin as to be expected. For 100 digraphs, the ROC curve does not begin at the origin because there are not always at least four or more of the same digraph present in testing sample that are also present in the reference users training sample. For this experiment, when our algorithm does not have enough information to make a decision it always accepts the test user. We believe the ROC curve does not begin at an equal true accept and false accept rate because when the test user is the reference user, it is more likely they have typed similar topics and words as contained in their training data. Whereas, imposters are less likely to have typed about the same topics and used the same words. However, because this is free-text keystroke dynamics and we are using enough data, the rates only differ by a few percent. If less data was used for training, it is possible the ROC curve would begin at a much higher false accept rate than true accept rate. As the amount of data used to for training increases we would expect to eventually have enough samples for every conceivable digraph and the ROC curve would begin at equal false accept and true accept rates.

Figure 5 shows the statistics of the digraphs from each of the different sizes of testing samples. Not surprisingly, as the testing sample size increases, more of the same digraphs occur with a higher frequency. With fewer digraphs in the testing sample, the probability of getting at least four of the same digraph is much lower than with a larger test sample. Additionally, if a digraph occurs four more times in the testing sample that digraph may not have four or more occurrences in the reference users training data. This is due to only 9000 digraphs present in the training sample. More data, while likely to eliminate this effect, is impractical to collect in a real time system as the training phase would be far too time consuming. If no digraphs can be compared, the test user is assumed to be the reference user causing the ROC curve with 100 digraphs not to start at the origin.

3.4. Results with less than 500 testing digraphs

Figures 6 and 7 compare the performance of the three metrics individually, fused in pairs, and fused all together, with 100 and 200 digraphs in the testing sample. Both fu-

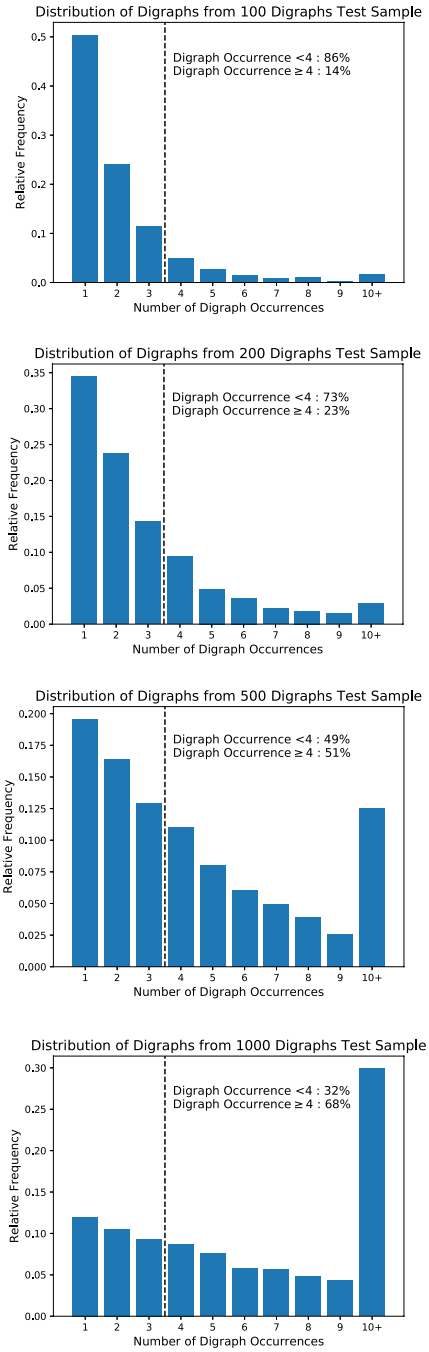

Figure 5. Frequencies of repeated digraphs from 100, 200, 500, and 1000 digraphs in the testing sample. The dashed black line distinguishes between digraphs that occur less than 4 times and digraphs that occur 4 or more times. Having $\geq$4 digraphs allows the algorithms to compute a CDF. As the testing sample size increases it becomes more likely the testing sample will contain a digraph that occurs 4 or more times.

sion metrics described in Section 2.4 are used for fusing the three algorithms. The KDE and Energy algorithms perform

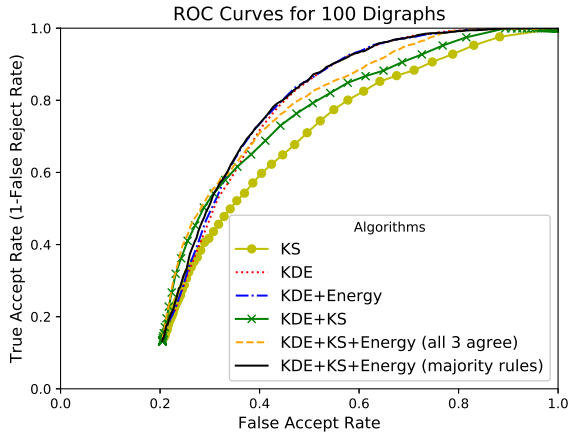

Figure 6. ROC curves for the KDE and KS metrics used for classification individually, fused in pairs, fused together with all three metrics authenticating, and fused all together with majority rules, with 100 digraphs in the testing sample. Their EER's are 35.8%, 40.3%, 37.0% (KDE and KS), 35.1% (KDE and Energy), 35.9% (all 3 authenticate), and 35.3% (majority rules) respectively.

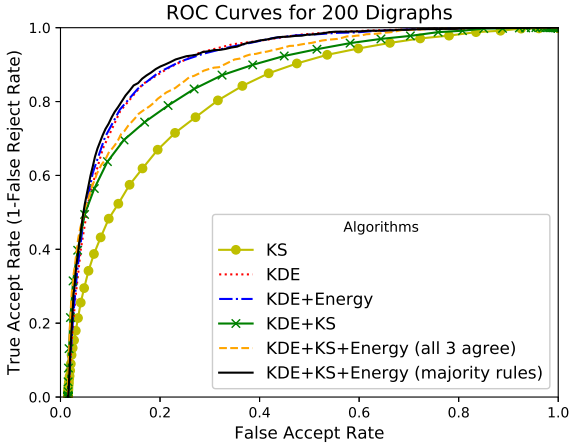

Figure 7. ROC curves for the KDE and KS metrics used for classification individually, fused in pairs, fused together with all three metrics authenticating, and fused all together with at least two of three authenticating, with 200 digraphs in the testing sample. Their EER's are 15.9%, 25.7%, 21.3% (KDE and KS), 16.0% (KDE and Energy), 19.4% (all 3 authenticate), and 15.3% (majority rules) respectively.

almost identically, as seen in Figure 4, and do not benefit much from fusing with each other. For testing samples of both 100 and 200 digraphs, fusing KS with KDE results in a lower EER that KDE alone.

For 100 digraphs in the testing sample, the ROC curves show that KDE fused with KS leads to higher true positive rates at lower false accept rates and lower true positive rates at higher false accept rates. The majority rules fu-

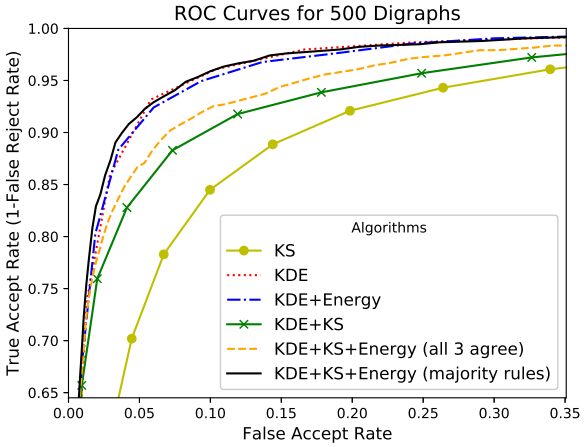

Figure 8. ROC curves for the KDE, and KS metrics used for classification individually, fused in pairs, and fused all together, with 500 digraphs in the testing sample. Their EER's are 6.3%, 12.8%, 10.1% (KDE and KS), 6.8% (KDE and Energy), 8.7% (all 3 authenticate), and 6.6% ($\geq$2 of 3 authenticate) respectively.

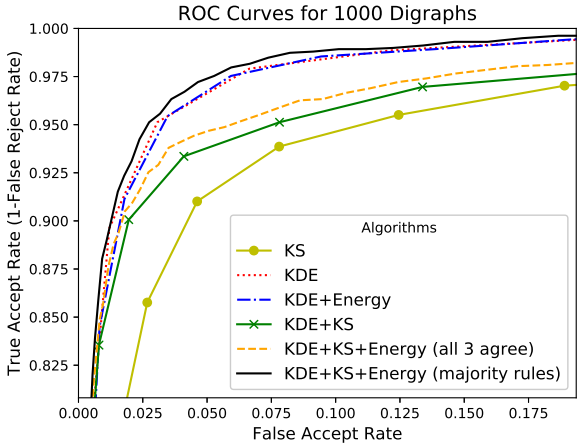

Figure 9. ROC curves for the KDE, KS, and Energy metrics used for classification individually, fused in pairs, and fused all together, with 1000 digraphs in the testing sample. Their EER's are 4.0%, 7.0%, 5.4% (KDE and KS), 4.0% (KDE and Energy), 5.2% (all 3 authenticate), and 3.6% ($\geq$2 of 3 authenticate) respectively.

sion classifier, requiring at least two of the three metrics to authenticate the test user, outperforms KS combined with KDE. Finally, the fused classifier requiring all three metrics to authenticate the test user performs best at low false accept rates, but is beaten by the majority rules fusion at higher false accept rates. This is expected since having all three metrics agree on authentication reduces the number of accepts, including false accepts. It should also be noted that the majority rules method is less strict on acceptance when compared to requiring all three metrics to authenticate. The majority rules method, therefore, is the best performer at higher true accept rates. It is a design consideration to select the fusion method to implement. For example, if security is desired (low false accept rates), then choosing the fusion method where all three metrics authenticate the test user is the better algorithm choice. Regardless of algorithm preference the EER with 100 digraphs in the testing sample will be around 35%.

For 200 digraphs in the testing sample, fusing KS with KDE yields worse overall performance than KDE individually. The fusion method requiring all three metrics to authenticate the test user also performs worse than KDE alone. Fusing all three metrics using majority rules outperforms KDE individually and is the best performing algorithm achieving an EER of 15.3%.

3.5. Results with greater than 500 testing digraphs

Figures 8 and 9 compare the performance of the three metrics individually, fused in pairs, and fused all together, with 500 and 1000 digraphs in the testing sample. As with the cases of less than 500 digraphs in the testing sample, the KDE and Energy algorithms perform similarly and do not benefit from fusing with each other. KS is the worst overall performing metric and fusing KS with KDE yields worse overall performance than KDE individually. The fusion method requiring all three metrics to authenticate the test user also performs worse than KDE alone.

For 500 digraphs in the testing sample, fusing all three metrics with majority rules provides the best overall performance at lower false accepts rates while at higher false accept rates, KDE individually performs very similarly. Majority rules fusion performs best, achieving an EER of 6.6%. Compared in [12], three state-of-the-art algorithms, KDE based [12], Gunetti and Picardi's [10], and Buffalo's SVM [5] algorithms, achieved EER's of 7.6%, 10.3%, and 15.7% respectively with 1000 keystrokes for the Clarkson II dataset. With 500 digraphs in the testing sample, our fused metric classifier achieves an EER just below the state-of-the-art systems. This demonstrates the ability of our classifier to quickly authenticate users, or authenticate users with half as many digraphs as in the state-of-the-art. For 1000 digraphs in the testing sample, fusing all three metrics with majority rules provides the best overall performance. An EER of 3.6% is achieved using the majority rules fusion method, which is a significant improvement over the state-of-the-art performance of an EER of 7.6% [12].

In Figure 10, we plot the EER for both of the fused distance metrics vs. the number of digraphs used for testing. With fewer then 80 samples, a stable EER cannot be calculated due to lack of data in the testing samples. Performance improves rapidly as we increase the number of digraphs from 80 to 300. As the number of digraphs continue to increase, performance improves, but slowly. For almost every amount of digraphs in the test sample, the fusion method re-

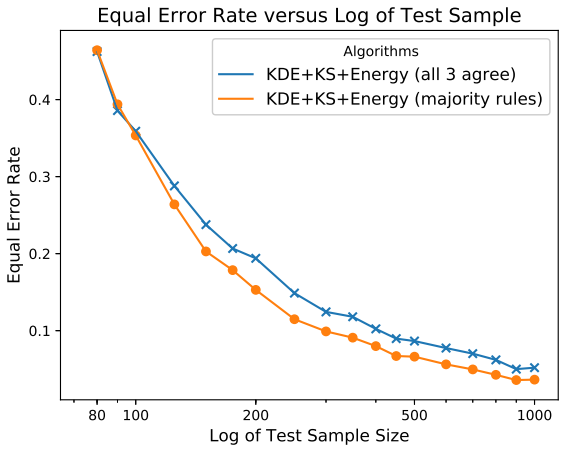

Figure 10. EER versus test sample size for fusion of the three algorithms with both fusion methods. For fewer than 80 digraphs, the performance is not reliable and an EER value was not generated. The fusion method where all three metrics are required to authenticate the test user appears to have a worse overall EER than the fusion method requiring only two of the three metrics to authenticate the test user for all testing samples with more than 100 digraphs.

quiring at least two of three metrics to authenticate the test user has a lower EER than the fusion method where all three metrics are required to authenticate the test user. With fewer than 100 digraphs in the testing sample, the EER is lower for the all three metric authentication fusion method. At 100 digraphs, we observed a trade-off between false accept rates and true positive rates for both fusion methods. Choosing the best performing algorithm for testing samples of size 100 or fewer, will depend on the desired system. For testing sample sizes of 100 or more digraphs the fusion method requiring at least two of the three metrics to authenticate the test user is the best performer.

4. Conclusions and Future Work

In this paper, we present three metrics for fast intruder detection using keystroke dynamics. In systems presented in literature, for good performance, around 1000 keystrokes are required. However, this leaves these systems vulnerable during short activities such as tweeting and composing short emails, about the length used for phishing attacks. To improve both the speed and frequency of authentication, we focused on testing samples with less than 1000 digraphs. We investigated the performance of kernel density estimation (KDE), Kolmogorov-Smirnov (KS), and Energy algorithms as individual metrics, fused in pairs, as well as all three fused. Table 2 shows EER's for the different fusion methods with 100, 200, 500, and 1000 digraphs in the testing samples. For the fused metrics, due to their ROC curves

Fusion Method	EER from # of test digraphs			
	100	200	500	1000
KS alone	40.3%	25.7%	12.8%	7.0%
KDE alone	35.8%	15.9%	**6.3%**	4.0%
KDE and KS	37.0%	21.3%	10.1%	5.4%
KDE and Energy	**35.1%**	16.0%	6.8%	4.0%
All 3 agree	35.9%	19.4%	8.7%	5.2%
Majority Rules	35.3%	**15.3%**	6.6%	**3.6%**

Table 2. EER's for the different fusion methods for 100, 200, 500, and 1000 digraphs in the testing sample. Due to the similar performance between the KDE and Energy metrics, Energy alone and Energy fused with KS are not reported. Majority rules is overall the strongest performing fusion method.

intersecting with 100 or fewer digraphs in the testing sample, to choose the better fused algorithm the EER should not be the only factor considered. The choice will also depend on whether the desired system should favor security or convenience. With only 100 digraphs in the testing sample the performance is not very strong. Our results improve with more digraphs in the testing sample.

In addition to our algorithm's promising performance for fast intruder detection, our fused classifier's performance is an improvement over existing state-of-the-art algorithms. With 1000 digraphs in the testing sample, the majority rules classifier achieves an EER of 3.6%. This is a significant improvement over other state-of-the-art algorithms with EER's of 7.6%, 10.3%, and 15.7% with 1000 digraphs in the testing sample [12]. With 500 digraphs in the testing sample the majority rules fusion classifier achieves an EER of 6.6%. Our classifier achieves slightly better performance over existing state-of-the-art methods with half as much data enabling faster and more frequent authentication. With 200 digraphs in the testing sample the EER rises to 15.3% with the majority rules fusion classifier. While the EER at 200 digraphs may be too large to be implemented as is, our continuous security layer can now authenticate with 20% of the data and 5 times as often when compared to previous state-of-the-art systems.

In this paper, two fusion methods were used: one where all metrics agree on a decision, and another where a majority two out of three metrics agree. Future work involves exploring other fusion methods. These include weighted fusion and optimizing fusion with neural networks and deep-learning. With 100 digraphs used for testing, there was not one ROC curve that was the best. Devising new metrics and fusions schemes that will provide improvements on the methods presented in this paper will also be investigated. Other avenues of future research include investigation of n-graphs for $n > 2$, and analyzing the effects of prior and future keystrokes to improve authentication models.

References

[1] M. Ali, J. Monaco, and C. Tappert. Hidden Markov models in keystroke dynamics. *Proceedings of student-faculty research day*, 2015. 1

[2] S. Banerjee and D. Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1):116–139, 2012. 2

[3] M. Bellemare, I. Danihelka, W. Dabney, S. Mohamed, B. Lakshminarayanan, S. Hoyer, and R. Munos. The Cramer distance as a solution to biased Wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017. 3

[4] M. Burnett. 10,000 top passwords. https://xato.net/10-000-top-passwords-6d6380716fe0. Accessed: 2019-4-15. 1

[5] H. Çeker and S. Upadhyaya. User authentication with keystroke dynamics in long-text data. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–6. IEEE, 2016. 7

[6] G. Corder and D. Foreman. *Nonparametric statistics: A step-by-step approach*. John Wiley & Sons, 2014. 3

[7] H. Davoudi and E. Kabir. A new distance measure for free text keystroke authentication. In *Computer Conference, 2009. CSICC 2009. 14th International CSI*, pages 570–575. IEEE, 2009. 1

[8] P. P. E. Jones, T. Oliphant et al. SciPy: Open source scientific tools for Python, 2001. 3

[9] K. Gessler. Stop mindlessly following character count recommendations on facebook posts, 2016. 1

[10] D. Gunetti and C. Picardi. Keystroke analysis of free text. *ACM Trans. Inf. Syst. Secur.*, 8(3):312–347, Aug. 2005. 2, 7

[11] J. Huang, D. Hou, S. Schuckers, and Z. Hou. Effect of data size on performance of free-text keystroke authentication. In *Identity, Security and Behavior Analysis (ISBA), 2015 IEEE International Conference on*, pages 1–7, March 2015. 2

[12] J. Huang, D. Hou, S. Schuckers, T. Law, and A. Sherwin. Benchmarking keystroke authentication algorithms. In *Information Forensics and Security (WIFS), 2017 IEEE Workshop on*, pages 1–6. IEEE, 2017. 1, 2, 5, 7, 8

[13] J. Huang, D. Hou, S. Schuckers, and S. Upadhyaya. Effects of text filtering on authentication performance of keystroke biometrics. In *WIFS*, pages 1–6, 2016. 2, 4

[14] F. J. Massey, Jr. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951. 3

[15] C. Murphy, J. Huang, D. Hou, and S. Schuckers. Shared dataset on natural human-computer interaction to support continuous authentication research. In *2017 IEEE International Joint Conference on Biometrics, IJCB 2017, Denver, CO, USA, October 1-4, 2017*, pages 525–530, 2017. 3

[16] M. Panzarino. Interesting fact: more tweets posted are 28 characters than any other length, 2012. 1

[17] K. I. Park. *Fundamentals of Probability and Stochastic Processes with Applications to Communications*. Springer, 2018. 2

[18] S. Park, J. Park, and S. Cho. User authentication based on keystroke analysis of long free texts with a reduced number of features. In *Communication Systems, Networks and Applications (ICCSNA), 2010 Second International Conference on*, volume 1, pages 433–435. IEEE, 2010. 2

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 3

[20] B. Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018. 2

[21] T. Sim and R. Janakiraman. Are digraphs good for free-text keystroke dynamics? In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2007. 2

[22] N. Solomon. The average sentence length, 2008. 1

[23] G. Székely. E-statistics: The energy of statistical samples. *Bowling Green State University, Department of Mathematics and Statistics Technical Report*, 3(05):1–18, 2003. 3

[24] P. Teh, A. Teoh, and S. Yue. A survey of keystroke dynamics biometrics. *The Scientific World Journal*, 2013. 1, 2, 4

[25] R. Yampolskiy and V. Govindaraju. Behavioural biometrics: a survey and classification. *International Journal of Biometrics*, 1(1):81–113, 2008. 1

[26] E. Yu and S. Cho. Keystroke dynamics identity verification-its problems and practical solutions. *Computers & Security*, 23(5):428–440, 2004. 1