Recent Advances in the Development of a Long-Text-Input Keystroke Biometric Authentication System for Arbitrary Text Input

John V. Monaco, Ned Bakelman, Sung-Hyuk Cha, and Charles C. Tappert
Seidenberg School of Computer Science and Information Systems
Pace University, White Plains, NY 10606, USA
{vinmonaco, nbakelman}@gmail.com, {scha, ctappert}@pace.edu

***Abstract*—This study focuses on the development and evaluation of a new classification algorithm that halves the previously reported best error rate. Using keystroke data from 119 users, closed system performance was obtained as a function of the number of keystrokes per sample. The applications of interest are authenticating online student test takers and computer users in security sensitive environments. The authentication process operates on keystroke data windows as short as ½ minute. Performance was obtained on 119 test users compared to the previous maximum of 30. For each population size, the performance increases, and the equal error rate decreases, as the number of keystrokes per sample increases. Performance on 14, 30, and 119 users was 99.6%, 98.3%, and 96.3%, respectively, on 755-keystroke samples, indicating the potential of this approach. Because the mean population performance does not give the complete picture, the varied performance over the population of users was analyzed.**

Keywords*—*pattern recognition, machine learning, biometrics, keystroke biometrics, user authentication, intruder detection

I. INTRODUCTION

This paper describes the recent advances in the development and evaluation of a keystroke biometric system for continual computer-user authentication on short-burst-input durations on the order of minutes. An application of this work is verifying the identity of students taking online tests, an application important for the 2008 United States Higher Education Opportunity Act which requires institutions of higher learning to make greater online access control efforts by adopting ubiquitous identification technologies [10]. Another is intruder detection – the discovery that somebody other than the authentic user is operating the computer [6]. The online student test-taking and intruder detection applications are similar in terms of authenticating the user, but faster discovery is required in the intruder case to prevent significant harm.

Keystroke biometric systems measure typing characteristics believed to be unique to an individual and difficult to duplicate [4, 11]. The keystroke biometric is one of the less-studied behavioral biometrics, usually relegated to conference sessions on "other biometrics" and described only briefly in books on biometrics. Nevertheless, the keystroke biometric has been reviewed in several recent articles [12, 21]. The keystroke biometric is appealing for several reasons. First, it is non-intrusive and transparent to computer users who type frequently for both work and pleasure. Second, it is inexpensive since the only hardware required is a computer with keyboard. Third, keystrokes continue to be entered for potential repeated verification after initial authentication since keystrokes exist as a mere consequence of using computers [9]. Continued verification throughout a computer session is known as dynamic verification [14] or active authentication [6]. Most of the earlier studies used passwords or short name strings [2, 4, 8, 13, 15, 18, 21, 22], and there are currently a number of commercial keystroke authentication products for password "hardening". A smaller number of earlier studies used long-text input as in this study [3, 9, 14, 16, 17, 19, 23, 25].

This study extends an earlier-used unique but not-well-known dichotomy model classification procedure, a strong inferential statistics method found to be particularly effective in large open biometric systems where it is not possible to train the system on all individuals in the population [5, 24]. The applications of interest here, however, involve closed populations where it is possible to train the system on all of the authorized users.

Therefore, a more accurate "engineering" procedure was developed for the closed-population applications. The new extension of the classification procedure is user-focused in that only the claimed user's enrollment samples and their relationship to the other users' enrollment samples are utilized in the classification process, in contrast to the previously employed procedure that uses all user relationships. Although focusing on one pattern class relative to the others is not new – for example, it is a hallmark of Support Vector Machines – this idea is an innovation in the context of the dichotomy classification model that operates on *differences of feature difference vectors*. The new procedure also matches the claimed user's sample against all the enrollment samples from that user for authentication rather than just one as in the previous system. Furthermore, using the leave-one-out method allows for the

evaluation of larger populations than reported in earlier studies [17, 25].

The performance of the improved system is evaluated on closed populations of 14, 30, and 119 users. In addition to the usual overall system performance evaluation by ROC curves, a breakdown of performance over the population of users describes the performance inhomogeneity.

The remaining sections of the paper present the methodology, the experimental results, the discussion, and the conclusions.

II. METHODOLOGY

This work continues the development of a keystroke biometric system to authenticate users of standard desktop/laptop computers. The system consists of a frontend application for data capture and feature extraction, and a backend one for authentication classification and ROC curve generation. An existing system was used for the frontend [23], while the backend classifier was improved and is the focus of this paper.

A. Data Capture

The keystroke data were captured in a Java applet that used the PC Windows-event clock that recorded the key press and release times in a millisecond format.

B. Feature Extraction

The feature extraction component extracts a vector of 239 features from the raw timing data. The features are statistical in nature and designed to characterize an individual's keystroke dynamics over writing samples of 200 or more characters. Most of the features are averages and standard deviations of key press duration times and of digraph transition times. While key press duration and transition times are typically used as features in keystroke biometric password authentication systems, our use of the statistical measures of means and standard deviations of the key presses and transitions is uncommon and only practical for long text input. As additional features, we use percentages of key presses of many of the special keys. Some of these percentage features are designed to capture the user's preferences for using certain keys or key groups – for example, some users do not capitalize or use much punctuation in email. The features are grouped as follows (see [23] for details):

- 78 duration features (39 means and 39 standard deviations) of individual letter and non-letter keys, and of groups of letter and non-letter keys
- 70 key-release-to-key-press transition features (35 means and 35 standard deviations) of the transitions between letters or groups of letters, between letters and non-letters or groups thereof, between non-letters and letters or groups thereof, and between non-letters and non-letters or groups thereof
- 70 key-press-to-key-press transition features (35 means and 35 standard deviations) identical to the above features except for the method of measurement
- 19 percentage features that measure the percentage of use of the non-letter keys and mouse clicks
- 2 keystroke input rates: the unadjusted input rate (total time to enter the text / total number of keystrokes and mouse events) and the adjusted input rate (total time to enter the text minus pauses greater than ½ second / total number of keystrokes and mouse events)

The computation of a keystroke-duration mean or standard deviation requires special handling when there are few samples. For example, when the number of samples for a keyboard key is less than a threshold, the mean is calculated as the weighted average of the mean of the key in question and the mean of the appropriate fallback group of keys at the next highest node in a hierarchy tree. Because we are dealing with long-text input, fallback is necessary for only infrequently used keys. Thus, we ensure computability (no zero divides) and obtain reasonable values for all feature measurements.

Two preprocessing steps are performed on the feature measurements: outlier removal and feature standardization. Outlier removal is particularly important for these features because a keyboard user could pause for a phone call, for a sip of coffee, or for numerous other reasons, and the resulting outliers – overly long transition times – would skew the feature measurements. Overly long key presses can also occur but are rare. Outlier removal consists of removing any duration or transition time that is more than two standard deviations from the mean values. After outlier removal, averages and standard deviations are recalculated recursively until no further outliers can be removed. After performing outlier removal, the feature measurements are standardized into the range 0-1 by clamping each measurement at plus and minus two standard deviations over all samples from all participants. This standardization method gives each measurement roughly equal weight. The feature measurements, the hierarchical trees, the fallback procedure, and the preprocessing steps have been described more fully in [23].

C. Authentication Classification

The classification procedure uses a vector-difference authentication model which transforms a multi-class problem into a two-class problem [5]. The resulting two classes are *within-person* ("you are authenticated") and *between-person* ("you are not authenticated").

To explain the dichotomy transformation process, consider a small example of three people $\{P_1, P_2, P_3\}$ where each person supplies four biometric samples. Fig. 1 plots the biometric sample data for these three people in two-dimensional feature space. This feature space is transformed into a feature-difference space by calculating vector distances between pairs of samples of the *same*

person (*within-person distances*, denoted by $x_{\oplus}$) and distances between pairs of samples of *different* people (*between-person distances*, denoted by $x_{\varnothing}$). Let d_{ij} represent the individual feature vector of the i^{th} person's j^{th} biometric sample, then the sets $x_{\oplus}$ and $x_{\varnothing}$ of vector differences are calculated as follows:

$$x_{\oplus} = \{|d_{ij} - d_{ik}| \text{ where } i=1 \text{ to } n, \text{ and } j,k=1 \text{ to } m, j \neq k\}$$
$$x_{\varnothing} = \{|d_{ij} - d_{kl}| \text{ where } i,k=1 \text{ to } n, i \neq k \text{ and } j,l=1 \text{ to } m\} \quad (1)$$

where n is the number of people, m is the number of samples per person, and the absolute value is of the elements of these vectors.

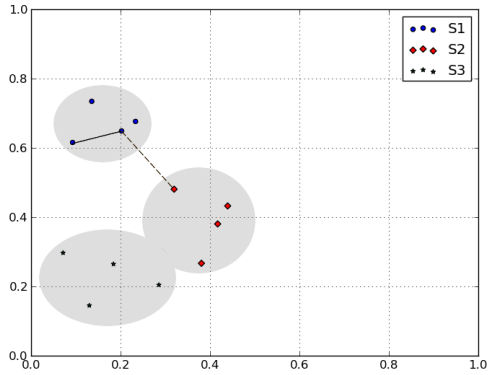

Fig. 1. Feature space: three subjects, four samples each.

If n people provide m biometric samples each, the numbers of within-person $n_{\oplus}$ and between-person $n_{\varnothing}$ distance samples, respectively, are:

$$n_{\oplus} = \frac{m \times (m-1) \times n}{2}, \quad n_{\varnothing} = m \times m \times \frac{n \times (n-1)}{2} \quad (2)$$

Fig. 2 shows the transformed feature difference space for the example, n = 3 and m = 4 yields $n_{\oplus}$ = 18 and $n_{\varnothing}$ = 48 for a total of 66 vector differences. The two highlighted difference samples come from the two lines in Fig. 1.

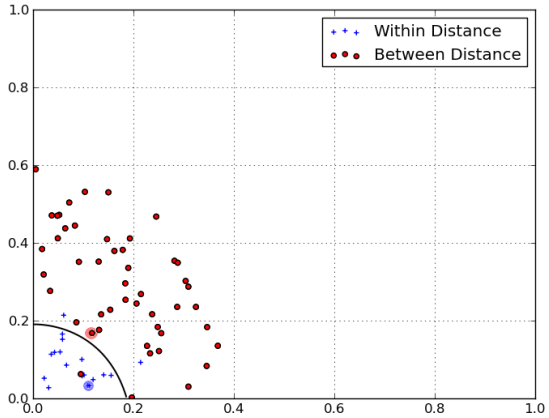

Fig. 2. Feature difference space, transformed from Fig. 1.

In the simulated authentication process, a claimed user's keystroke sample requiring authentication is first converted into a feature vector. The differences between this feature vector and all the earlier-obtained enrollment feature vectors from this user are computed, and the resulting difference vectors are matched against the within-person training difference vectors for authentication or between-person ones for non-authentication using the *k*-nearest-neighbor procedure. Thus, *differences of difference vectors are being calculated*. Because most pattern recognition systems calculate difference vectors in the matching/classification process, the fact that the dichotomy model takes differences of difference vectors is often not understood as different and unique.

The training space of difference vectors grows rapidly as the population increases, particularly the number of between-person distance samples – for example, 200 users each producing 10 enrollment samples generates 1 990 000 between-person distance samples. Thus, it is necessary to reduce the number of training difference vectors, and previously a random sampling was performed. Now, however, an improved reduction method has been discovered that has led to significantly higher performance.

For efficiency and performance the improved user-focused reduction method retains only training difference vectors that include the claimed user's test samples. Thus, the numbers of within-person $n_{\oplus}$ and between-person $n_{\varnothing}$ distance samples, respectively, become:

$$n_{\oplus} = m \times (m-1)/2, \quad n_{\varnothing} = m \times m \times (n-1) \quad (3)$$

For the small example illustrated above, Fig. 3 shows the corresponding feature-difference spaces for user S1, yielding $n_{\oplus}$ = 6 and $n_{\varnothing}$ = 32.

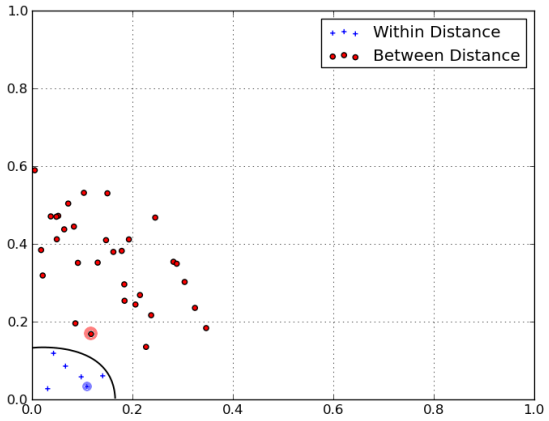

Fig. 3. Feature difference space for user S1.

For large n the number of vector difference samples, especially the between-person differences, is greatly

reduced. For example, in the above mentioned example of 200 users each producing 10 enrollment samples, the number of between-person distance samples is reduced from 1 990 000 to 19 900.

More importantly, this user-focused approach improves performance by taking into account the clustering of the individual user's samples. Fig. 1 shows three clusters of samples from the three users. Fig. 3 shows the feature difference space for only user S1 where the within-class feature-difference samples are clustered rather tightly, corresponding to the tight cluster for user S1 in feature space (Fig. 1). In contrast, the within-class feature-difference samples in Fig. 2 are less tightly clustered because they represent the feature-difference samples from all three users. Now, realizing that Fig. 3 characterizes this phenomenon for only two features to permit a two-dimensional drawing, consider the greater overall tighter clustering effect of this user-oriented approach in a higher-dimensional pattern feature space.

To obtain system performance the leave-one-out procedure simulates many true users trying to get authenticated and many imposters trying to get authenticated as other users. The leave-one-out method also allows for the evaluation of larger populations than has been reported in earlier studies [17, 25]. The procedure is implemented as follows. For each question ("left-out") test sample to be authenticated, the within and between-person difference vectors are computed without that sample. This creates the training space, which consists of the entire population less the sample to be authenticated. The testing vectors are then computed by taking the difference vectors between the question test sample and enrollment samples belonging to the claimed user. The results of a nearest neighbor classification for each of the test difference vectors are grouped together to make the decision by considering the nearest neighbors from all the resulting vector differences.

D. ROC Curve Derivation

Receiver operating characteristic (ROC) curves characterize the performance of a biometric system and show the trade-off between the False Accept Rate (FAR) and the False Reject Rate (FRR). In this study, the ROC curves were obtained by using a weighted procedure of the k nearest neighbors [25]. This procedure uses a linear rank weighting, assigning the first choice (nearest neighbor) a weight of k, second a weight of $k-1$, ... , and the k^{th} a weight of 1. The maximum score when all k choices are within-person is $k+(k-1)+...+1 = k(k+1)/2$, and the minimum score is 0. Now, consider that we authenticate a user if the weighted-within-person choices are greater or equal to l, where l varies from 0 to $k(k+1)/2$, and compute the (FRR, FAR) pairs for each l to obtain an ROC curve. The Equal Error Rate (EER) is where FAR = FRR on the ROC curve. The experiments below use k = 21 to provide weighted scores in the range 0-231 and thus 232 points on the ROC curve. This value of k was chosen to generate a reasonable number of points on the ROC curves. When deploying the system the value of l is chosen to establish an appropriate operating point trade-off between FAR and FRR on the ROC curve.

III. EXPERIMENTAL RESULTS

These experiments employed free-text (arbitrary input) keystroke data samples from an earlier study [17]. All the data samples contained over 500 keystrokes, averaged 755 keystrokes, and were input on Dell desktop PCs and laptop PCs (almost exclusively Dell machines). The data samples were collected in sets of five, the sets recorded at two-week intervals, and the five samples of a set usually recorded in a single day's session. For their five samples in a set, the participants were instructed to enter emails on five different topics from a given list of topics. Data were collected from 120 participants but one produced only two samples and was eliminated from the study.

Three experiments were conducted to analyze performance as a function of the number of keystrokes per sample and the user population size (Table 1).

Table 1. Summary of experimental data and EER for the 755-keystroke samples.

Exp	Number of Participants	Number of Samples Each	Total Number of Samples	EER(%)
1	14	15	210	0.4
2	30	10	300	1.7
3	119	5	595	3.7

Each of the experiments involved positive and negative authentication tests – the number of positive tests = *number-of-samples* and the number of negative tests = *number-of-samples times (n-1)*. For example, for the 119 participant experiments, the 595 keystroke samples allowed for the evaluation of 595 positive and 70 210 (595x118) negative tests. The negative tests were zero-effort imitations by other subjects in the database.

Figs. 4, 5, and 6 show the key results for the 14, 30, and 119 user experiments, respectively. Each of these figures shows the ROC curves on the left and the FAR/FRR versus the l (L) parameter for the 755-keystroke samples on the right. Although the EER can be approximated from the ROC curve, it can be more accurately determined form the crossover point on the FAR/FRR versus L curve (note that because the lowercase l can be confused with the digit 1 we sometimes use the uppercase L to represent the parameter). Although L goes from 0-231, expanded FAR/FRR plots at low L values are shown here because the crossover points on the FAR/FRR versus L curves occur in that region.

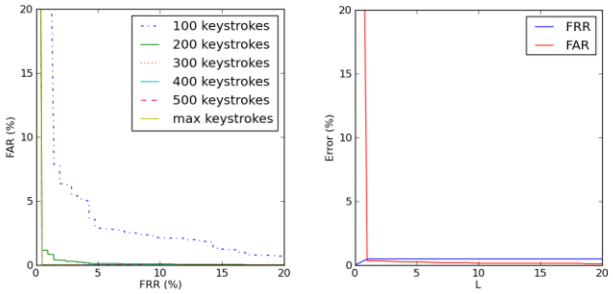

Fig. 4. Experiment 1: 14 users, ROC curves (left) and FAR/FRR versus *L* for max keystrokes (right).

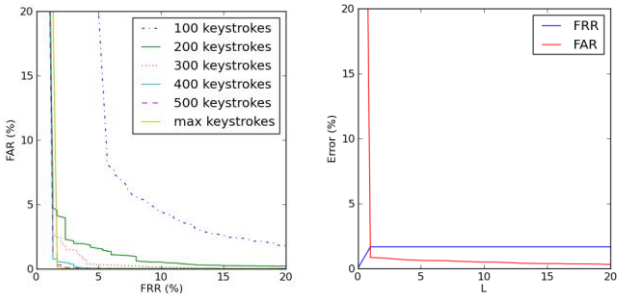

Fig. 5. 30 users ROC curves (left) and FAR/FRR versus *L* for max keystrokes (right).

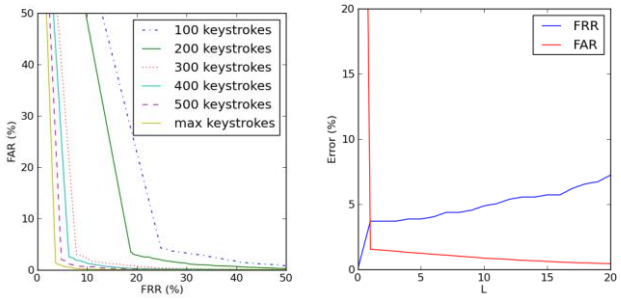

Fig. 6. 119 users ROC curves (left) and FAR/FRR versus *L* for max keystrokes (right).

To determine how fast an unauthorized user could be detected, the EER rate was determined as a function of sample length (number of input keystrokes) for the 14, 30, and 119 user populations (Fig. 7).

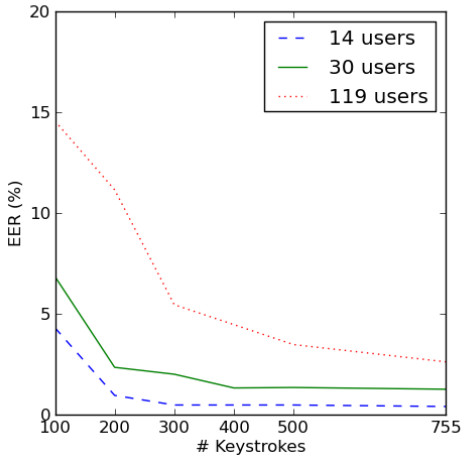

Fig. 7. EER versus #keystrokes.

For the maximum length keystroke samples, Fig. 8 shows the ROC curves. The EERs were 0.4%, 1.7%, and 3.7% for the 14, 30, and 119 participant populations, respectively.

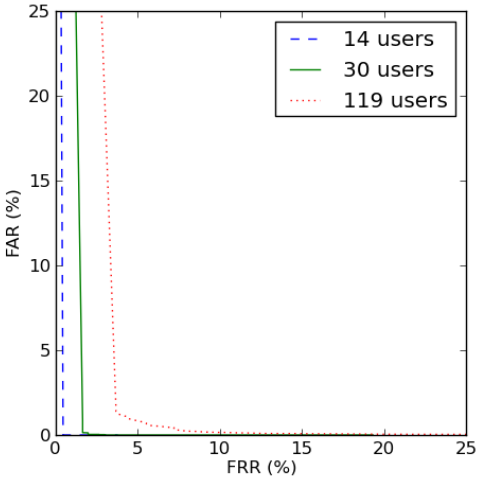

Fig. 8. ROC curves for 14, 30, and 119 users for the maximum keystroke samples.

Because the mean population performance does not give the complete picture, the varied performance over the population of users was analyzed and described using the animal designations of Doddington et al. [7]. Figs. 9 and 10 show three histograms analyzing the populations of 30 and 119 users for the maximum keystroke samples when operating at the EER point on the ROC curve. For each population size the three histograms show the FRR (potentially identifying those easily verified, *sheep*, and those difficult to verify, *goats*), the FAR of how easily the true authors were imitated (potentially identifying those easily attacked, *lambs*), and the FAR of how easily imitators can attack the true users (potentially identifying the strong attackers, *wolves*).

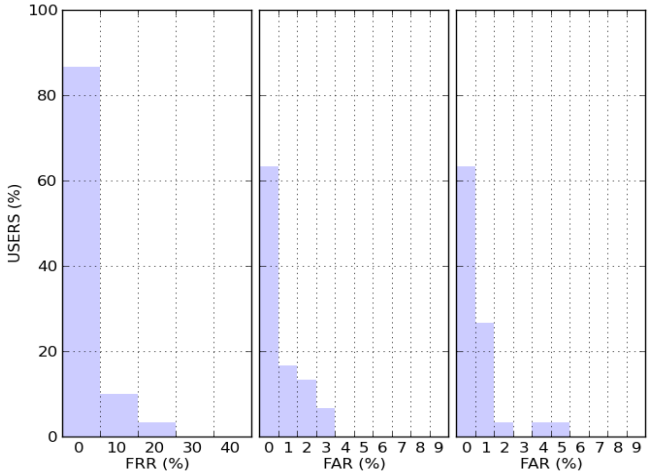

Fig. 9. Histograms of FRR (left), FAR of attack receivers (middle), and FAR of attackers (right) over the 30 user population.

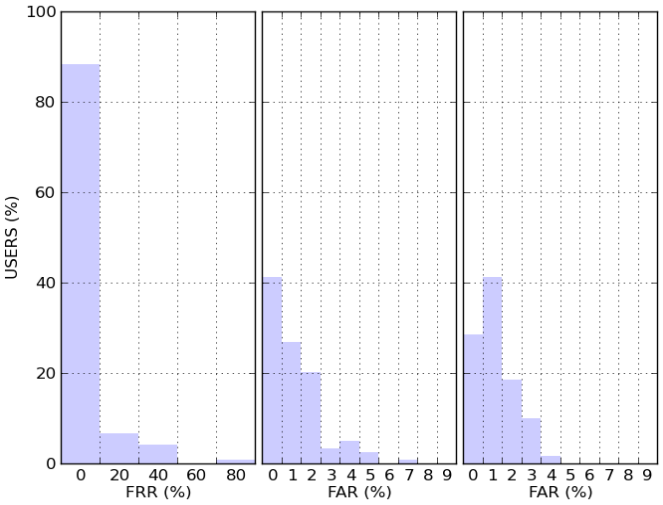

Fig. 10. Histograms of FRR (left), FAR of attack receivers (middle), and FAR of attackers (right) over the 119 user population.

An examination of these histograms found only one significant outlier and that occurred in the FRR histogram of the 119 user population. This FRR histogram displays the percent of the 119 users (5 samples each) having 0% (0 of 5), 20% (1 of 5), 40% (2 of 5), 60% (3 of 5), 80% (4 of 5), and 100% (5 of 5) false rejects. Because one user had 80% (4 of 5) samples rejected while all others had 40% or fewer, this user is an outlier and could be considered a *goat* (a user difficult to authenticate).

IV. Discussion

To obtain system performance in this study we simulated the authentication process of many true users trying to get authenticated and of many zero-effort imposters trying to get authenticated as other users. An important advantage of this vector-difference model is that it provides relatively large numbers of between- and within-person distance samples for analysis and ROC curve generation. Furthermore, the leave-one-out method allowed for the closed-system evaluation of a considerably larger population size than had been evaluated previously.

As in a study by Bartmann et al. [1], the approach taken in this study was to train on as much enrolment data as possible while authenticating users on smaller quantities of data as appropriate for the application. For the test taker application there is usually no hurry to authenticate the user and all of the keystroke data for an online test could be used for authentication. However, for detecting unauthorized users in security sensitive applications the quantity of keystroke data must be limited to a minute or so in order to detect the intruder before significant harm is committed. Because large quantities of training data from authentic users – perhaps over many days, weeks, and even months – are available for some applications, such as the intruder detection application, elaborate procedures for training the system on significant quantities of data should be investigated.

For using such a continual authentication system on government or private company machines, keylogger software could be installed to transparently capture user input on all monitored PCs and the authentication processing performed on servers. However, because many employees like to use their PC for occasional personal use – email, banking, stock market transactions, etc. – there are obvious privacy concerns with a keylogger capturing all input, including account numbers and passwords. And, although the organizations might say they can monitor their machines as they like, the employees could have strong objections. To increase user acceptance and ameliorate privacy concerns, monitored machines should be clearly marked as such and unmonitored machines could be made available for employee personal use during lunch and break times. Although privacy concerns remain, for authenticating test takers this should not be a problem because the students would be using a test-taking application.

For the intruder detection application, it is important to relate typing speed to the number of keystrokes entered per minute, and it will be done in this discussion for the English language. The average word length is five, plus a space, or six characters per word. For average computer users, the average typing speed is about 33 words per minute, while a professional typist's speed is about twice that of the average user, and keyboard key spacing has an effect on the typing speed [20]. Since the number of keystrokes is usually only slightly more than the number of characters, the average computer user generates about 200 keystrokes per minute, while a professional typist generates about 400 keystrokes per minute. Assuming an average typing speed, 1-2 minutes of a potential intruder's input would likely be in the 200-400 keystroke range. In examining the error rate as a function of the number of keystrokes (Fig. 7), a big drop occurs in going from 100 to 200 keystrokes for 14 and 30 users, and in going from 200 to 300 keystrokes for 119 users, which implies that at least 200-300 keystrokes, or 1 to 1½ minutes,

is necessary to detect an intruder, which is hopefully fast enough to stop the intruder from causing damage.

V. Conclusions

The main contributions of this study were the development of the improved classification system and its performance evaluation. On samples of 300 or more keystrokes (1½ minutes or more at average typing speed) performance was over 98% on 30 users and over 94% on 119 users. On the large 755-keystroke samples performance reached 98.3% on 30 users and 96.3% on 119 users, indicating the potential of this approach.

In this study the EER was used for simplicity as a single value of performance to show the trends of performance as a function of the number of keystrokes per sample and the population size. However, in setting up the procedure for authenticating a user in a deployed system, the operating point on the ROC curve would be chosen appropriately, usually with a considerably lower FAR than FRR. For example, a good operating point on the ROC curves for 14 and 30 users in Fig. 8 might be FRR = 2% and FAR ~ 0%. Although a low FAR operating point would incur more false rejections, several authentication failures could be required before making an unauthorized-user decision. How easily users accept such a system and at what value of FRR it becomes too intrusive are problems for future work concerning system deployment, maintenance, and user acceptance.

Acknowledgment

We thank the student teams in the masters level projects course that contributed to this effort over recent years.

References

[1] D. Bartmann, I. Bakdi, and M. Achatz, "On the design of an authentication system based on keystroke dynamics using a predefined input text," *Int. J. Info. Security and Privacy*, vol. 1, 2007.

[2] S. Bender and H. Postley, "Key sequence rhythm recognition system and method," *US Patent 7,206,938*, 2007.

[3] F. Bergadano, D. Gunetti, and C. Picardi, "User authentication through keystroke dynamics," *ACM Trans. Info. Sys.*, 2002, pp.367-397.

[4] R. Bolle, J. Connell, S. Pankanti, N. Ratha, and A. Senior, *Guide to biometrics*. New York: Springer, 2004.

[5] S. Cha and S. Srihari, "Writer identification: statistical analysis and dichotomizer," in *Advances in Pattern Recognition*. vol. 1876: Springer, 2000, pp.123-132.

[6] DARPA. (Apr 2012). *Active Authentication Program*. https://www.fbo.gov/index?s=opportunity&mode=form&id=c7968647352f0276fc1b28817c581d86&tab=core&_cview=0

[7] G. Doddington, W. Liggett, A. Martin, M. Przybocki, and D. Reynolds, "Sheep, goats, lambs, and wolves: A statistical analysis of speaker performance. *Proc. IC-SLD'98, NIST 1998 speaker recognition evaluation*, Sydney, Australia, 1998.

[8] R. Giot, M. El-Abed, and C. Rosenberger, "Keystroke dynamics with low constraints svm based passphrase enrollment," *IEEE Int. Conf. Biometrics (BTAS 2009)*, 2009.

[9] D. Gunetti and C. Picardi, "Keystroke analysis of free text," *ACM Trans. Info. Sys.*, vol. 8, 2005, pp.312-347.

[10] HEOA. U.S. *2008 Higher Education Opportunity Act (HEOA)*. www2.ed.gov/policy/highered/leg/hea08/index.html, May 2011.

[11] L. Jin, X. Ke, R. Manual, and M. Wilkerson, "Keystroke dynamics: A software based biometric solution," *13th USENIX Sec. Sym.*, 2004.

[12] M. Karnan, M. Akila, and N. Krishnaraj, "Biometric personal authentication using keystroke dynamics: A review," *Applied Soft Computing J.*, vol. 11, 2011.

[13] K. Killourhy and R. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," *Int. Conf. Dependable Systems & Networks (DSN-09)*, Lisbon, 2009, pp. 125-134.

[14] F. Leggett, G. Williams, and M. Usnick, "Dynamic identity verification keystroke characteristics," *Int. J. Man-Machine Studies*, 1991, pp. 859-870.

[15] Y. Li, B. Zhang, Y. Cao, S. Zhao, Y. Gao, and J. Liu, "Study on the BeiHang keystroke dynamics database," *Int. Joint Conf. Biometrics (IJCB 2011)*, Washington, D.C., 2011.

[16] A. Messerman, C. Mustafi, S. Camtepe, and S.Albayrak, "Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics," *Int. Joint Conf. Biometrics (IJCB 2011)*, Washington D.C., 2011.

[17] J.V. Monaco, N. Bakelman, S. Cha, and C.C. Tappert, "Developing a keystroke biometric system for continual authentication of computer users," *Proc. Euro. Intell. Sec. Inform. Conf.*, Denmark, 2012, pp. 210-216.

[18] F. Monrose, M. Reiter, and S. Wetzel, "Password hardening based on keystroke dynamics," *Int. J. Info. Security*, vol. 1, 2002, pp. 69-83.

[19] A. Peacock, X. Ke, and M. Wilkerson, "Typing patterns: A key to user identification," *IEEE Security & Privacy*, vol. 2, 2004, pp. 40-47.

[20] A. Pereira, D.L. Lee, H. Sadeeshkumar, C. Laroche, D. Odell, and D. Rempel, "The effect of keyboard key spacing on typing speed, error, usability, and biomechanics: Part 1," *J. Human Factors and Ergonomics Soc.*, Nov 2012.

[21] K. Revett, "Chapter 4: Keystroke dynamics," in *Behavioral biometrics: A remote access approach*: Wiley, 2008, pp. 73-136.

[22] R. Rodrigues, G. Yared, C. Costa, J. Yabu-Uti, F. Violaro, and L. Ling, "Biometric access control through numerical keyboards based on keystroke dynamics," in *Lecture Notes in Computer Science*. vol. 3832, 2005, pp. 640-646.

[23] C. Tappert, S. Cha, M. Villani, and R. Zack, "A keystroke biometric system for long-text input," *Int. J. Info. Sec. Privacy*, 2010, pp. 32-60.

[24] S. Yoon, S.Choi, S. Cha, Y. Lee, and C. Tappert, "On the individuality of the iris biometric," *Int. J. Graphics, Vision and Image Processing*, 2005, pp. 63-70.

[25] R. Zack, C. Tappert, and S.Cha, "Performance of a long-text-input keystroke biometric authentication system using an improved k-nearest-neighbor classification method," *IEEE 4th Int. Conf. Biometrics (BTAS 2010)*, Washington D.C., 2010.