Utilizing Keystroke Dynamics as Additional Security Measure to Protect Account Recovery Mechanism

Ahmed Anu Wahab[1][a], Daqing Hou[1][b], Stephanie Schuckers[1][c] and Abbie Barbir[2]

[1] *Electrical and Computer Engineering, Clarkson University, Potsdam NY, U.S.A.*

[2] *Mobile Security Group, CVS Health, U.S.A.*

Keywords: Behavioral Biometric, Keystroke Dynamics, Free-text, Fixed-text, Forgot Password/Username, Account Recovery.

Abstract: Account recovery is ubiquitous across web applications but circumvents the username/password-based login step. Therefore, it deserves the same level of security as the user authentication process. A common simplistic procedure for account recovery requires that a user enters the same email used during registration, to which a password recovery link or a new username could be sent. Therefore, an impostor with access to a user's registration email and other credentials can trigger an account recovery session to take over the user's account. To prevent such attacks, beyond validating the email and other credentials entered by the user, our proposed recovery method utilizes keystroke dynamics to further secure the account recovery mechanism. Keystroke dynamics is a type of behavioral biometrics that uses the analysis of typing rhythm for user authentication. Using a new dataset with over 500,000 keystrokes collected from 44 students and university staff when they fill out an account recovery web form of multiple fields, we have evaluated the performance of five scoring algorithms on individual fields as well as feature-level fusion and weighted-score fusion. We achieve the best EER of 5.47% when keystroke dynamics from individual fields are used, 0% for a feature-level fusion of five fields, and 0% for a weighted-score fusion of seven fields. Our work represents a new kind of keystroke dynamics that we would like to call it 'medium fixed-text' as it sits between the conventional (short) fixed text and (long) free text research.

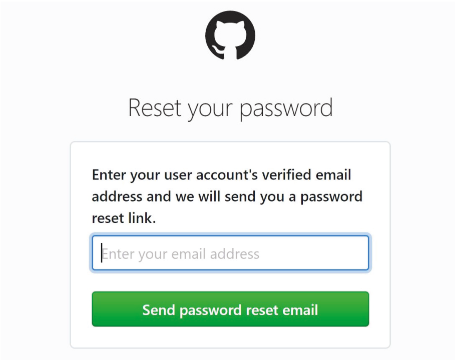
1 INTRODUCTION

The username and password have been the dominant means of verifying a user's digital identity over the years, but also fraught with many security problems. For example, in the first half of 2018 alone, it was estimated that about 4.5 billion online user accounts were exposed, a majority of which as a result of password breaches (Gemalto Inc, 2018). Because of the difficulty in remembering passwords, many users have been known to use a single password across multiple websites, making it easier for an impostor to take over their accounts. To increase security, a common practice has been adopted by many sites to require users to regularly change their passwords and to use long unique passwords, for example, as a combination of uppercase and lowercase alphabets, numbers, and symbols. Consequently, many users find it even harder to remember passwords. These challenges with username and password necessarily popularize the account recovery mechanism on the web. Figure 1 shows a common recovery method that simply sends a recovery link to a user's verified email. While this is appropriate for sites with low security requirements, to increase the level of security, many sites also require the user to perform additional verification, such as answering security questions or providing personal credentials (Figure 2). However, it is also well known that security questions and personal information can be stolen through social engineering or brute-force attacks.

Perhaps the most dangerous vulnerability that the account recovery mechanism can lead to is the fact that any impostor with access to a user's recovery email (which can be taken over by attacks such as credential stuffing (owasp.org, 2020)) can easily trigger an account recovery session and take over the user's account. Given that account recovery is ubiquitous across the web and being widely

[a] https://orcid.org/0000-0003-4677-5269
[b] https://orcid.org/0000-0001-8401-7157
[c] https://orcid.org/0000-0002-9365-9642

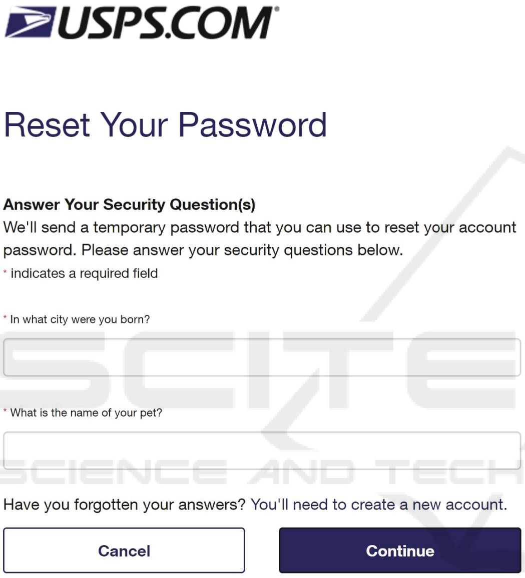
Wahab, A., Hou, D., Schuckers, S. and Barbir, A.
Utilizing Keystroke Dynamics as Additional Security Measure to Protect Account Recovery Mechanism.
DOI: 10.5220/0010191200330042
In *Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP 2021)*, pages 33-42
ISBN: 978-989-758-491-6



Reset your password

Enter your user account's verified email address and we will send you a password reset link.

Figure 1: Github password recovery requires a user's verified email address to send the password reset link.



Reset Your Password

Answer Your Security Question(s)

We'll send a temporary password that you can use to reset your account password. Please answer your security questions below.

* indicates a required field

* In what city were you born?

* What is the name of your pet?

Have you forgotten your answers? You'll need to create a new account.

Figure 2: As additional protection, United State Postal Service (USPS) account recovery also requires a user to answer security questions.

used by enterprise information systems, it deserves the same level of security as the user authentication process. To that end, we propose to verify a user's identity through behavioral biometrics using the keystroke dynamics collected during the password/username recovery session. Research has demonstrated that keystroke dynamics can be a useful behavioral biometrics for authentication (Rybnik et al., 2009)(Choraś and Mroczkowski, 2007)(Revett et al., 2006) but does not require additional hardware. Our research goal is to further strengthen the security of the account recovery mechanism using keystroke dynamics. We envision that this modality can be fused with other modalities to form a more robust risk-based scoring system to ensure that the person requesting account recovery is indeed the claimed user.

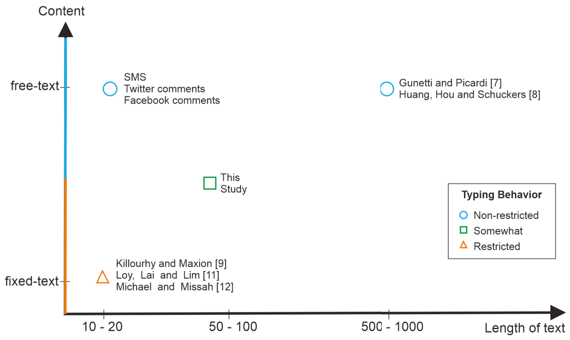


Figure 3: Characterizing keystroke dynamics based on three traits: Length of text (long or short), typing behaviour (restricted or unrestricted) and typed content (fixed or free). Our study is between fixed-text and free-text in a laboratory setting (somewhat restricted).

In this paper we focus on sites that have implemented additional verification during account recovery by requesting more information from the user. Using an account recovery form with multiple fields, we have collected a new dataset with over 500,000 keystrokes from 44 students and staff of our university. We investigate the authentication performance of keystroke dynamics from both individual fields and their various combinations. We implement five state-of-the-art scoring algorithms for both fixed-text and free-text keystroke dynamics to measure the similarity between the test samples and the established user profile. These algorithms either accept or reject the user based on the returned score and a threshold on the score. We achieve the best EER of 5.47% when using individual fields, and 0% for both a feature-level fusion and a weighted-score fusion.

As shown in Figure 3, work on keystroke dynamics can be characterized by length (short or long), typing behavior (restricted or unrestricted contexts) and typed content (fixed or free/varied across sessions). When unrestricted, users type anything on their own regular device at any time and anywhere of their choice. Fixed text (also known as static text) refers to cases when the text needed to perform keystroke analysis is constant during enrollment and testing. An example of a short length fixed-text in keystroke dynamics is password, where users are required to type a password with fixed and unchanging characters. Free text (also known as dynamic text) refers to cases when users are allowed to type freely with no constraint on when/where/what to type. An example of a long length free-text is when a user writes an article on a topic of their own interest. When keystrokes from each field in our dataset is used individually for authentication, this work can be considered as short length, fixed-text keystroke dynamics; but when fields in the dataset are combined into a long text, then our work can be consider as free-text. Therefore, this study sits somewhere in the middle of fixed-text and free-text, and we would like to call it ‘medium length, fixed-text.’ Note also that the medium fixed-text keystroke dynamics has put little to none restrictions on our users’ typing behavior other than the fact that they type in our laboratory.

The remaining of this paper is organized as follows. Section 2 presents related work in both fixed-text and free-text keystroke dynamics. Section 3 describes our methodology: the dataset, feature extraction, algorithms and implementation procedure. Results and findings are presented in Section 4. Lastly, Section 5 concludes the paper.

2 LITERATURE REVIEW

Keystroke dynamics is the analysis of typing rhythm which can be used for authentication. It involves inspecting timing features of an individual’s typing and latency between keys to identify patterns in the keystroke data. In the eighties, Gaines et al. (Gaines et al., 1980) investigated whether individuals could be distinguished in the ways they type, by examining the probability distributions of the times each typist typed pairs of successive letters (digraphs) while typing a paragraph of prose. Since then, researchers have come up with many more applications and techniques for keystroke dynamics (Banerjee and Woodard, 2012), (Teh et al., 2013), (Alsultan and Warwick, 2013).

Gunetti and Picardi (Gunetti and Picardi, 2005) is among the first exploring free text keystroke dynamics using digraphs, the latencies between two successive keystrokes, which have been commonly used in short (fixed) text research. Their work on free-text shows that relatively long text samples with about 800 characters are required to accurately differentiate between a genuine user and impostors. Huang et al. (Huang et al., 2015) finds that in free-text, larger reference profiles with more digraphs will drive down both impostor pass rate (IPR) and false alarm rate (FAR), provided that the test samples have sufficient digraphs, but more digraphs in test samples beyond 1000 seem to have no obvious effect on IPR, regardless of the size of the reference profile. Generally, test samples of 500 to 1000 digraph instances have been used in free-text literature (Figure 3). In this regard, our work is unique because it is not completely free-text or fixed-text, but somewhere in between. Our work has achieved better accuracy with fewer digraph instances than Gunetti and Picardi (Gunetti and Picardi, 2005) and Huang et al. (Huang et al., 2015).

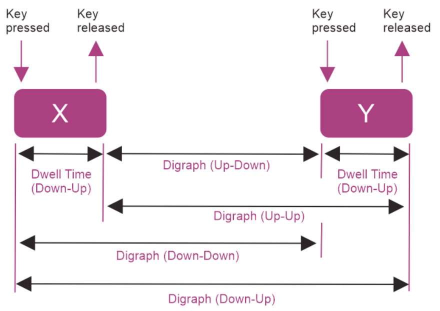


Figure 4: Keystroke dynamics features (dwell/hold time and digraph latency defined in terms of key press/release events).

Keystroke dynamic features are extracted by using the timing information of keys pressed, which includes latency between consecutive keys and dwell/hold time of a single key. As shown in Figure 4, the latency between keys may include the time interval between the press of a key and the press of the next key (down-down), the interval between the release of a key and the press of the next key (up-down) or interval between the release of a key and the release of the next key (up-up). The dwell/hold time is the interval between the press and the release of a single key (down-up). Many studies have been done on fixed-text keystroke dynamics for password (Pisani and Lorena, 2013), (Revett et al., 2006), (Monrose et al., 2002), (Bartlow and Cukic, 2006), (de Magalhaes et al., 2005) and free-text (Gunetti and Picardi, 2005), (Huang et al., 2017), but ours is the first study on the use of keystroke dynamics to further protect account recovery mechanism.

Many keystroke dynamics datasets for password impose the same fixed password string for all users such as Killourhy and Maxion (Killourhy and Maxion, 2009), Loy, Lai and Lim (Loy et al., 2007), and Michael and Missah (Michael and Missah, 2016). Killourhy and Maxion have a dataset of 20,400 samples, collected from 51 subjects and each subject contributed 400 typing samples of the same string “.tie5Roanl”. Out of the 14 recognition algorithms used in their work, they report Scaled Manhattan, Nearest Neighbor (Mahalanobis) and Outlier Count as the best three performing recognition algorithms with EER of 9.6%, 10% and 10.2% respectively. However, an imposed password is unrealistic, because when users use their actual passwords, performance may vary. To investigate this possible difference in performance, Giot, El-Abed and Rosenberger (Giot et al., 2012) create a dataset with samples collected from 83 users (Table 1), a total of 5,185 genuine samples (pair of chosen username and password typed by its owner), 5,754 impostor samples (pair of username

Table 1: Password datasets for keystroke dynamics.

Dataset	#Users	#Samples	User Specific Password?
Killourhy and Maxion (Killourhy and Maxion, 2009)	51	20,400	No
Giot, El-Abed and Rosenberger (Giot et al., 2012)	83	5,185+ 5,754/5,439	Yes
BioChaves (Montalvao et al., 2006)	47	1,400	No
Allen (Allen, 2010)	104	2,736	No
Keystroke100 (Loy et al., 2007)	100	1,000	No
GREYC-NISLAB (Idrus et al., 2013)	110	2,201	No

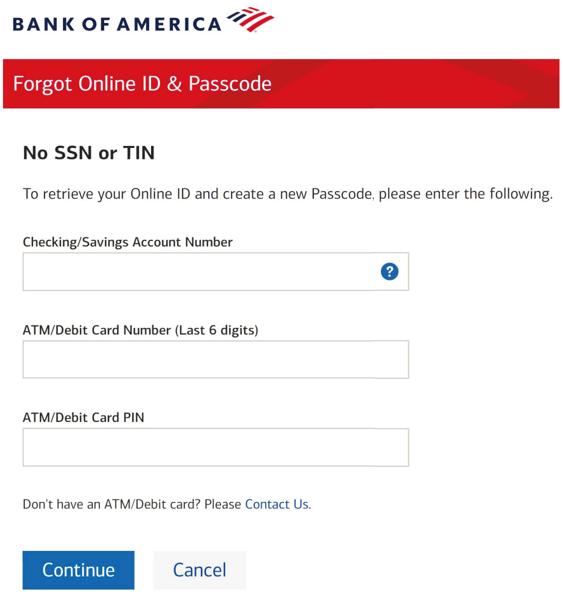
and password typed by a user different of its owner), and 5,439 imposed samples (pair of imposed username and password). Although their work seems to be realistic to real user scenario of different password selection, they find a surprising result that there is no significant difference in performance between the chosen and the imposed datasets. They had claimed that a possible explanation is, even though users were asked to choose a password of their own, they did not choose their real password and would have chosen a password they are less familiar with. They have also reported an issue with quality measure during data collection which could have been the cause for their underlined surprising observation. In contrast, our work in account recovery is based on a practical and realistic scenario.

3 EXPERIMENTAL DESIGN

The account recovery mechanisms implemented on many public and business websites collect either a single field (a registration email address) or multiple fields of information (e.g., email, phone number, address, and full name) from users. The required number of fields to trigger an account recovery session is related to the level of security of the platform and the value placed on the account. For example, while the Github website requires just a single email address (Figure 1), an online banking platform, which is more security-sensitive, would request multiple fields of information for added security (Figure 5). For improved security, we have collected multiple fields of information from users during our data collection.

3.1 Account-Recovery Keystroke Dataset

We have created a new Account-Recovery dataset with a total of over 500,000 keystrokes. The data was collected from 44 university students and staff using a data collection web app (Figure 6). Each user visits us twice. In the first visit, each user fills an enrollment form on the web app ten times. The keystrokes collected from the enrollment form are used to build the user's profile. In the second visit one or two weeks later, each user fills the form again five times, which is used as the user's genuine keystrokes. The same user also attacks five other users each twice, which serves as impostor keystrokes. As a result, our new dataset contains data for when users attack each other. Figure 7 depicts such an example where user (ID: W0037-81456) attacks another user W0092-17843.



BANK OF AMERICA

Forgot Online ID & Passcode

No SSN or TIN

To retrieve your Online ID and create a new Passcode, please enter the following.

Checking/Savings Account Number

ATM/Debit Card Number (Last 6 digits)

ATM/Debit Card PIN

Don't have an ATM/Debit card? Please Contact Us.

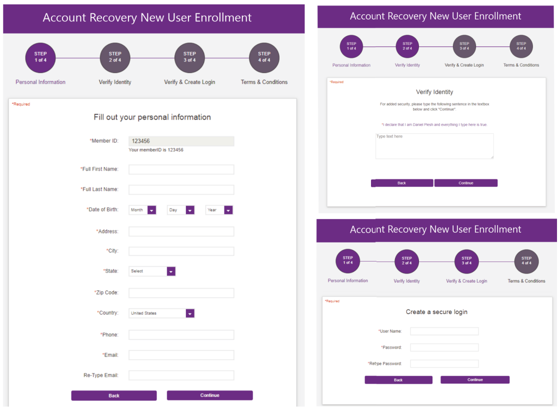
Figure 5: Bank of America forget password session requires a user to enter multiple fields of information.

The enrollment form consists of the following fields: *Full name, Address, City, Zip, Phone, Email, Declaration, and Password*. Users are asked to type the following text as *declaration*: *"I declare that I am (Full name) and everything I type here is true"* (also see Figure 6). The dataset holds the record of key-down and key-up timing information of every key pressed and released, and our users are allowed to make and correct typing errors.

Overall, 42 users complete the enrollment process ten times as requested (the other two complete less than ten times). 28 users return in a second visit to fill the enrollment form for five more times, but only 16 of the 28 have acted as imposters.

3.2 Data Preprocessing and Cleaning

Since our dataset allows for typing errors, we preprocess the raw data to remove backspaces and the keystrokes deleted by the backspaces, which may have been used for correcting misspellings. Table 2 and 3 show a summary of keys contributed per user and per field, respectively, after data cleaning and preprocessing.


Account Recovery New User Enrollment

STEP 1 of 4 Personal Information — STEP 2 of 4 Verify Identity — STEP 3 of 4 Verify & Create Login — STEP 4 of 4 Terms & Conditions

*Required

Fill out your personal information

*Member ID: 123456
Your memberID is 123456
*Full First Name:
*Full Last Name:
*Date of Birth: Month Day Year
*Address:
*City:
*State: Select
*Zip Code:
*Country: United States
*Phone:
*Email:
Re-Type Email:

Back Continue

Account Recovery New User Enrollment

STEP 1 of 4 Personal Information — STEP 2 of 4 Verify Identity — STEP 3 of 4 Verify & Create Login — STEP 4 of 4 Terms & Conditions

*Required

Verify Identity

For added security, please type the following sentence in the textbox below and click "Continue".

*I declare that I am Daniel Plesh and everything I type here is true.

Type text here

Back Continue

Account Recovery New User Enrollment

STEP 1 of 4 Personal Information — STEP 2 of 4 Verify Identity — STEP 3 of 4 Verify & Create Login — STEP 4 of 4 Terms & Conditions

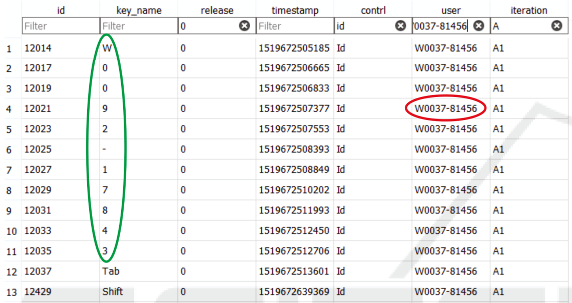
*Required

Create a secure login

*User Name:
*Password:
*Retype Password:

Back Continue

Figure 6: Account Recovery dataset: User interfaces of the data collection web app.



	id	key_name	release	timestamp	contrl	user	iteration
	Filter	Filter	0	Filter	id	0037-81456	A
1	12014	W	0	1519672505185	Id	W0037-81456	A1
2	12017	0	0	1519672506665	Id	W0037-81456	A1
3	12019	0	0	1519672506833	Id	W0037-81456	A1
4	12021	9	0	1519672507377	Id	W0037-81456	A1
5	12023	2	0	1519672507553	Id	W0037-81456	A1
6	12025	-	0	1519672508393	Id	W0037-81456	A1
7	12027	1	0	1519672508849	Id	W0037-81456	A1
8	12029	7	0	1519672510202	Id	W0037-81456	A1
9	12031	8	0	1519672511993	Id	W0037-81456	A1
10	12033	4	0	1519672512450	Id	W0037-81456	A1
11	12035	3	0	1519672512706	Id	W0037-81456	A1
12	12037	Tab	0	1519672513601	Id	W0037-81456	A1
13	12429	Shift	0	1519672639369	Id	W0037-81456	A1

Figure 7: Account Recovery dataset: User W0037-81456 attacks the profile of user W0092-17843.

We observe some inaccuracies and inconsistencies in the password field as many users did not use their true passwords or used them inconsistently across sessions. Such password data would not give meaningful information about the user's typing patterns. As a result, we do not use the password field. Similar user behavior has been noted elsewhere (Giot et al., 2012).

Table 2: Keystrokes per user after data pre-processing.

	Avg / Min / Max keys per User	#User
Profile	2,048 / 1,282 / 3,510	42
Genuine	1,210 / 614 / 3,219	28
Impostor	2,351 / 88 / 7,615	16

Table 3: Keystrokes per field after data pre-processing.

Fields	Avg / Min / Max keys per Field
Full name	13 / 4 / 20
Address	17 / 8 / 38
City	9 / 5 / 17
Zip	6 / 5 / 10
Phone	12 / 10 / 26
Email	21 / 15 / 37
Declare Text	68 / 53 / 135

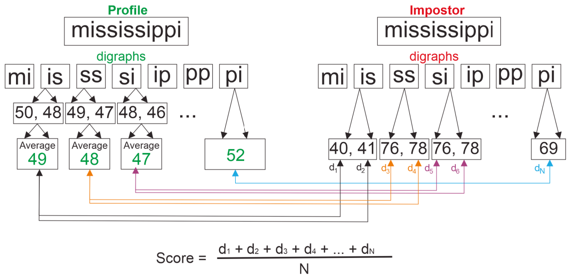


Figure 8: Scoring procedure for sample text 'mississippi' where s_i represents the timing difference between the *ith* digraph in the test sample and the profile.

3.3 Scoring Algorithms

We have implemented five state-of-the-art scoring algorithms from both fixed text and free text keystroke dynamics (Gunetti and Picardi, 2005), (Huang et al., 2017), (Killourhy and Maxion, 2010), (Killourhy and Maxion, 2009): Euclidean Distance, Manhattan Distance, Scaled Manhattan Distance, Mahalanobis Distance, and the 'A and R' Measures of Gunetti and Picardi (Gunetti and Picardi, 2005).

Figure 8 illustrates how the scoring algorithms work. Note that in the sample text 'mississippi', the digraph 'is', 'ss' and 'si' are repeated twice, while digraph 'mi', 'ip', 'pp' and 'pi' occur only once, making a total of seven unique digraphs. We calculate the average timing of the digraphs that have two instances (repeated twice) in the profile sample as shown in Figure 8. For each digraph instance in the test sample, our scoring algorithms compute the difference $(d_1, d_2, ..., d_N)$ between its timing and the timing of the same digraph in the profile. The overall distance score is the average of all individual differences, which measure how dissimilar the test sample is to the user profile. The higher the distance score, the less likely the test sample keystrokes belong to the user and vice-versa. In our implementation, we discard all digraphs that are longer than $\frac{1}{2}$ of a second. Such digraphs are typically the results of a user taking a break after making a typing error or pausing to attend to other tasks, and are less likely to be informative; the resulting time information would be an outlier and would negatively affect performance.

3.3.1 Euclidean Distance

Euclidean distance is the straight-line distance between two points in Euclidean space, which is calculated as follows:

$$D = \sqrt{\sum_{i=1}^{N} (\mu_{g_i} - x_i)^2}, \qquad (1)$$

where N is the number of digraphs shared between the test sample and the profile, x_i is the individual test graph duration for the i^{th} shared graph in the test sample, and μ_{g_i} is the mean of the i^{th} graph in the profile.

3.3.2 Manhattan Distance

The scaled Manhattan and Manhattan distance metrics were used by Kilhourhy and Maxion for fixed-text keystroke dynamics (Killourhy and Maxion, 2009). The scaled Manhattan distance is calculated as follows:

$$D = \sum_{i=1}^{N} \frac{|\mu_{g_i} - x_i|}{\sigma_{g_i}}, \qquad (2)$$

where N is the number of digraphs shared between the test sample and the profile, x_i is the individual test graph duration for the i^{th} shared graph in the test sample, and μ_{g_i} and σ_{g_i} are the mean and standard deviation of the i^{th} graph in the profile (Killourhy and Maxion, 2009). The Manhattan and scaled Manhattan distances are identical, except the Manhattan distance is not divided by the standard deviation (Black, 2019).

3.3.3 Mahalanobis Distance

The Mahalanobis distance is similar to the scaled Manhattan distance and is given by:

$$D = \sqrt{\sum_{i=1}^{N} \frac{(\mu_{g_i} - x_i)^2}{\sigma_{g_i}^2}}, \qquad (3)$$

where N is the number of digraphs shared between the test sample and the profile, x_i is the individual test graph duration for the i^{th} shared graph in the test sample, and μ_{g_i} and σ_{g_i} are the mean and standard deviation of the i^{th} graph in the profile (Killourhy and Maxion, 2009) and (Mahalanobis, 1936).

3.3.4 Gunetti and Picardi's Metric

Gunetti and Picardi's free-text algorithm (Gunetti and Picardi, 2005) combines typing speed (A-measure) and the degree of disorder (R-measure) to measure similarity (Huang et al., 2017). The 'A' measure represents the distance between typing samples S1 and S2 in terms of n-graphs (that is, n consecutive keystrokes; n=2 in our case), as follows:

$$A_{\mathrm{t,n}}(S1, S2) = 1 - \frac{\#similar}{\#shared}$$

where t is a constant for determining n-graph similarity. For example, let $\mathrm{G}_{\mathrm{S1,L1}}$ and $\mathrm{G}_{\mathrm{S2,L2}}$ be the same n-graph occurring in typing samples S1 and S2, with latencies L1 and L2, respectively. We say that $\mathrm{G}_{\mathrm{S1,L1}}$ and $\mathrm{G}_{\mathrm{S2,L2}}$ are similar if and only if $1 \leq \max(\mathrm{L1,L2})/\min(\mathrm{L1,L2}) \leq t$. The 'R' measure on the other hand quantifies the degree of disorder between two sequences M and M′, as the sum of the differences between the respective ranks of each element in M and M′.

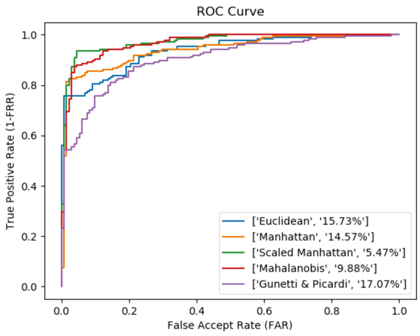


Figure 9: Receiver Operating Characteristics (ROC) curve for all five algorithms based on the *Declare* field, with Scaled Manhattan Distance being the best (EER of 5.47%).

3.4 Experiments

Consistent with the state-of-the-art in fixed-text keystroke dynamics (Killourhy and Maxion, 2009), as shown in Figure 9, Scaled Manhattan Distance outperforms the other four algorithms on the *Declare* field. Table 4 shows further evidence that this is also true for most of the remaining six fields. Therefore, further experiments in this paper are done in terms of the Scaled Manhattan Distance.

To identify fields and their combinations that produce the best authentication performance, we have performed several experiments to evaluate both individual fields and their fusions at both the feature and score levels. The result of each experiment is presented and discussed in Section 4.

Our data collection has allowed for some flexibility in the degree of content matching between data in the user profile and the test samples. This gives us the freedom to deploy a quality control mechanism K, which is the percentage of exact content matching between the profile and the test sample. We use K as a threshold to determine if a test sample will be included in our experiments or not. We have used three values for K (70%, 80% and 90%) in each experiment and recorded the K that produces the lowest EER.

3.4.1 Feature-level Fusion

This experiment evaluates the fusion of fields at the feature level. Our goal is to find the combination of fields that gives the highest accuracy (the lowest

Table 4: Performance of scoring algorithms on individual fields. Scaled Manhattan Distance is the overall best.

Field	Euclidean Distance EER (%)	Manhattan Distance EER (%)	Scaled Manhattan Distance EER (%)	Mahalanobis Distance EER (%)	Gunetti & Picardi Distance EER (%)
Zip	25.33	25.20	22.80	**21.84**	28.69
City	**19.51**	19.52	20.36	20.85	26.88
Phone	22.41	18.25	**18.02**	22.50	39.59
Fullname	17.29	16.31	**14.16**	16.04	20.67
Address	15.41	13.63	**10.81**	10.96	18.17
Email	12.59	9.62	**8.10**	12.45	15.75
Declare	15.73	15.74	**5.47**	9.88	17.07

EER). Specifically, we merge all the keystrokes from multiple fields and apply the Scaled Manhattan Distance scoring algorithm. We have carried out six major combinations which we named *Duet* (combination of two fields), *Trio* (combination of three fields), *Quartet* (combination of four fields), *Quintet* (combination of five fields), *Sextet* (combination of six fields) and *Septet* (combination of seven fields).

3.4.2 Weighted-Score Fusion

This experiment evaluates the weighted score fusion, where the final score D is defined as a weighted sum of individual field scores d_i ($D = w_1 \times d_1 + w_2 \times d_2 + ... + w_N \times d_N$), and all weights sum up to one ($w_1 + w_2 + ... + w_N = 1$). We use the grid-search approach to find the optimum weights for each combination. The grid-search approach is known to perform well for finding optimum weights in behavioral biometrics (Sitová et al., 2015).

3.4.3 Minimum Number of Enrollment Samples

In keystroke dynamics, enough enrollment samples are required to build the user's profile. The more the enrollment samples included in a user's profile, the more accurate the algorithm will perform. Although there is not a definite number of enrollment samples required to build a good profile, we have monitored performance as we reduce the number of enrollment samples. During our data collection, users have completed the enrollment process ten times and we have used all ten enrollment samples to build their profile. However, to further investigate the minimum number of enrollment samples, we experiment with varying the number of enrollment samples from 10 to 5 using both feature-level and weighted score fusion techniques.

4 RESULTS

This section presents the result for each experiment, including individual fields, feature level fusion and score level fusion.

Table 5: Authentication based on individual fields.

Field	#Avg shared digraph	K	#Comparison	EER (%)
Zip	4	90%	175	22.80
City	7	70%	343	20.36
Phone	8	70%	191	18.02
Fullname	12	70%	311	14.16
Address	16	70%	277	10.81
Email	20	70%	333	8.10
Declare	51	70%	304	5.47

Table 6: Feature level fusion of multiple fields.

Field	#Shared digraph	K	#Comparison	EER (%)
DUET				
Email+Fullname	29	90%	165	4.88
TRIO				
Declare+Email +Address	78	70%	254	3.13
QUARTET				
Declare+Email+ Address+Fullname	82	70%	224	2.36
QUINTET				
Declare+Email +Address+ Fullname+City	90	90%	52	0.00
SEXTET				
Declare+Email +Address+Fullname+ City+Zip	95	90%	48	0.00
SEPTET				
Declare+Email +Address+Fullname+ City+Zip+Phone	102	70%	227	2.18

4.1 Result for Individual Fields

Table 5 shows the performance of the Scaled Manhattan Distance over the seven fields on our account recovery web form. 'Declare', 'Email', and 'Address' are the three best performing fields with EER of 5.47%, 8.1%, and 10.81%, and an average of 51, 20, and 16 digraphs, respectively. The 'Zip' field has the lowest accuracy with EER of 22.8%, with a very short average of only 4 digraphs. As shown, field lengths seem to greatly influence performance and likely to be the main reason why the 'Declare' field has the best performance. On the other hand, familiarity with text may also have a relatively strong influence on performance. This is because more familiar content, such as email, are more likely to reveal a user's typing pattern.

Table 7: Weighted-Score fusion of multiple fields.

Field	#Comparison	K	EER (%)
PAIR			
Email(w=0.75)+Declare(w=0.25)	293	70%	4.3
TRIO			
Email(w=0.55)+Declare(w=0.25) +Fullname(w=0.2)	257	70%	2.7
QUARTET			
Email(w=0.45)+Declare(w=0.25)+ Fullname(w=0.15)+Address(w=0.15)	231	70%	2.27
QUINTET			
Email(w=0.45)+Declare(w=0.25) +Fullname(w=0.1)+ Address(w=0.15)+Zip(w=0.05)	201	70%	2.21
SEXTET			
Email(w=0.4)+Declare(w=0.25) +Fullname(w=0.1)+Address(w=0.1)+ Zip(w=0.1)+Phone(w=0.05)	141	70%	1.4
SEPT			
Email(w=0.35)+Declare(w=0.25) +Fullname(w=0.15)+Address(w=0.05)+ Zip(w=0.05)+Phone(w=0.05)+City(w=0.01)	83	80%	0.00

4.2 Result for Feature-level Fusion

Out of the seven fields in our account recovery form, there are 21 combinations for Duet (two fields), 35 combinations for Trio (three fields), 35 combinations for Quartet (four fields), 21 combinations for Quintet (five fields), 7 combinations of Sextet (six fields) and 1 for Septet (seven fields). Table 6 depicts the best performance for each of the above field combinations.

Consistent with the observed impact of the length of text on accuracy, an overall trend in the table is that as the number of shared digraph increases, EER decreases. We achieve 0% EER at the combination of five fields (Quintet) with an average of 90 shared digraphs and a *K* of 90%. Therefore, we do not need to fuse all seven fields to achieve perfect accuracy. Furthermore, we observe that the best field combinations in Table 6, from Trio down to Sextet, are mostly made of the set of best individual fields from Table 5. For example, the best combination of fields in Quartet is Declare+Email+Address+Fullname, which are the four best fields. However, we notice a performance drop at Septet (a combination of seven fields) despite an increase in the average shared digraph. Future work needs investigate the cause of this.

4.3 Result for Weighted-score Fusion

As recorded in Table 7, the global best result for weighted-score fusion is achieved at the combination of seven fields with EER of 0% and 83 comparisons. Consistent with the observed positive impact of the length of text on accuracy, an overall trend is that as the number of shared digraph increases, EER decreases. Furthermore, compared with the feature-level fusion, the weighted-score fusion performs better for Duet, Trio and Quartet, with lower EERs and higher number of comparisons. Overall, we believe the weighted score-level fusion is a better choice for our application because it uses more data (number of comparisons) and produces better performances when a less restrict content matching is applied (*K* is 70%; *K* is 80% for the combination of seven fields (Septet)).

Table 8: Number of enrollment samples and their corresponding EER values using feature-level fusion.

Field	Number of enrollment samples					
	10	9	8	7	6	5
DUET						
Email+Fullname	4.88	8.86	9.19	9.47	11.89	10.64
TRIO						
Declare+Email +Address	3.13	3.96	5.55	7.72	6.46	7.09
QUARTET						
Declare+Email+ Address+Fullname	2.36	3.17	4.74	5.37	8.39	10.44
QUINTET						
Declare+Email +Address+ Fullname+City	0.00	0.00	2.00	1.85	5.38	8.31
SEXTET						
Declare+Email +Address+Fullname+ City+Zip	0.00	0.00	0.00	0.00	5.65	7.98
SEPTET						
Declare+Email +Address+Fullname+ City+Zip+Phone	2.18	3.58	3.22	3.36	4.04	6.91

Table 9: Number of enrollment samples and their corresponding EER values using weighted-score fusion.

Field	Number of enrollment samples					
	10	9	8	7	6	5
DUET						
Email+Declare	4.3	4.37	4.46	5.8	5.83	6.04
TRIO						
Email+Declare +Fullname	2.7	3.87	3.47	4.26	5.29	5.3
QUARTET						
Email+Declare+ Fullname+Address	2.27	3.09	2.61	3.88	4.97	4.92
QUINTET						
Email+Declare +Fullname+ Address+Zip	2.21	3.49	3.04	3.97	4.85	4.81
SEXTET						
Email+Declare +Fullname+Address+ Zip+City	1.4	0.90	1.78	3.52	3.83	4.55
SEPTET						
Email+Declare +Fullname+Address+ Zip+City+Phone	0.00	0.00	1.88	3.54	3.85	3.58

These results outperform the state-of-the-art in both fixed-text and free-text keystroke dynamics. The best EER performance recorded in fixed-text papers like Killourhy and Maxion (Killourhy and Maxion, 2009), and Giot, EL-Abed and Rosenberger (Giot et al., 2012) are 9.6%, and 8.87% EER respectively, but we have achieved the lowest EER of 5.47% for individual fields. Likewise, we have achieved a global best EER of 0% for both feature-level fusion and weighted-score fusion, which outperform the results recorded in free-text papers like Gunetti and Picardi (Gunetti and Picardi, 2005) and Huang et al. (Huang et al., 2017) (Huang et al., 2015).

4.4 Result for Number of Enrollment Samples

Table 8 and Table 9 show the results of our experiment on the minimum number of enrollment samples using the feature-level fusion and weighted score fusion respectively. In general performance drops (i.e EER increases) as we reduce the number of enrollment samples. Furthermore, as the combination of fields increases, the reduction in the number of enrollment samples has a lesser effect on performance. For example, in Table 8, for the combination of five fields (Quintet), when the enrollment sample is reduced from 10 to 9, the performance stays the same (0%) but degrades when the enrollment sample is further reduced to 8. Meanwhile, for the combination of six fields (Sextet), performance stays the same as 0% when enrollment samples reduce gradually from 10 till 7. A possible explanation is, as fields are combined, the total number of digraphs increases, which counters the negative effect from the reduction in enrollment samples. Hence, short test samples would require more enrollment samples to build a user profile than long text in order to accomplish the same level of accuracy.

5 CONCLUSIONS

We propose to utilize keystroke dynamics as an additional security measure to further protect the account recovery mechanism. To that end, we have evaluated five scoring algorithms on our new account recovery dataset and find Scaled Manhattan Distance to be the best. We achieve the best EER of 5.47% when using individual fields, a global best EER of 0% with five fields combined using feature-level fusion and 0% for weighted-score fusion with all seven fields combined. In deciding the number of enrollment samples needed to build a user's profile, we find that a short test sample would require more enrollment samples than a long test sample. Overall, our results outperform the state-of-the-art in both fixed-text and free-text keystroke dynamics.

Keystroke dynamics provides an opportunity to reduce friction during Multi-factor Authentication (MFA) and ultimately improves users experience while providing additional security. Although there are possibilities of inconsistent keystrokes due to cramped muscles or sweaty hands, in such cases, keystroke dynamics would rightfully reject the users. Furthermore, the problem could be solved by requesting the user to present other Multi-factor Authentication (MFA) for authentication. It is important to stress that other MFA such as one-time password (OTP) inconveniences users and increases authentication friction. Keystroke dynamics can be used to significantly reduce such friction by requesting other MFA only when the user is rightfully rejected, such as in the cases of cramped muscles or sweaty hands.

Future work includes replicating this work on larger datasets with more keystrokes per user and more users, performing usability testing with real users, and the fusion with other modalities to form a more robust risk-based scoring system to ensure that the person requesting account recovery is indeed the claimed user. Also, typing patterns may vary on different keyboards. we are currently collecting data to study the variation of keystroke dynamics on different keyboards.

ACKNOWLEDGMENTS

This material is based upon work supported by the Center for Identification Technology Research (CITeR) and the National Science Foundation under NSF Grant No. 1650503, NSF Grant No. 1314792, and a grant from the NYSTAR Technology Transfer program (Contract #C180035). Dr. Jiaju Huang designed and collected the original data and performed a preliminary performance investigation in 2018.

REFERENCES

Allen, J. D. (2010). *An analysis of pressure-based keystroke dynamics algorithms*. PhD thesis, Southern Methodist University.

Alsultan, A. and Warwick, K. (2013). Keystroke dynamics authentication: a survey of free-text methods. *International Journal of Computer Science Issues (IJCSI)*, 10(4):1.

Banerjee, S. and Woodard, D. (2012). Biometric authentication and identification using keystroke dynamics:

A survey. *Journal of Pattern Recognition Research*, 7(1):116–139.

Bartlow, N. and Cukic, B. (2006). Evaluating the reliability of credential hardening through keystroke dynamics. In *2006 17th International Symposium on Software Reliability Engineering*, pages 117–126. IEEE.

Black, P. E. (2019). Manhattan distance. *Available online at: https://www.nist.gov/dads/HTML/ manhattanDistance.html. Last Accessed: 2019-06-15*.

Choraś, M. and Mroczkowski, P. (2007). Keystroke dynamics for biometrics identification. In *International Conference on Adaptive and Natural Computing Algorithms*, pages 424–431. Springer.

de Magalhaes, S. T., Revett, K., and Santos, H. M. (2005). Password secured sites-stepping forward with keystroke dynamics. In *International Conference on Next Generation Web Services Practices (NWeSP'05)*, pages 6–pp. IEEE.

Gaines, R. S., Lisowski, W., Press, S. J., and Shapiro, N. (1980). Authentication by keystroke timing: Some preliminary results. Technical report, Rand Corp Santa Monica CA.

Gemalto Inc (2018). Analysis: data breaches compromised 4.5bn records in half year 2018. https://thecitizenng.com/analysis-data-breaches-compromised-4-5bn-records-in-half-year-2018-gemalto/. Accessed: 2019-09-20.

Giot, R., El-Abed, M., and Rosenberger, C. (2012). Web-based benchmark for keystroke dynamics biometric systems: A statistical analysis. In *2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 11–15. IEEE.

Gunetti, D. and Picardi, C. (2005). Keystroke analysis of free text. *ACM Transactions on Information and System Security (TISSEC)*, 8(3):312–347.

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

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

Idrus, S. Z. S., Cherrier, E., Rosenberger, C., and Bours, P. (2013). Soft biometrics database: A benchmark for keystroke dynamics biometric systems. In *2013 International Conference of the BIOSIG Special Interest Group (BIOSIG)*, pages 1–8. IEEE.

Killourhy, K. and Maxion, R. (2010). Why did my detector do that?! In *International Workshop on Recent Advances in Intrusion Detection*, pages 256–276. Springer.

Killourhy, K. S. and Maxion, R. A. (2009). Comparing anomaly-detection algorithms for keystroke dynamics. In *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, pages 125–134. IEEE.

Loy, C. C., Lai, W. K., and Lim, C. P. (2007). Keystroke patterns classification using the artmap-fd neural network. In *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, volume 1, pages 61–64. IEEE.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. National Institute of Science of India.

Michael, O. B. and Missah, Y. M. (2016). Utilizing keystroke dynamics as an additional security measure to password security in computer web-based applications-a case study of uew. *International Journal of Computer Applications*, 149(5):35–44.

Monrose, F., Reiter, M. K., and Wetzel, S. (2002). Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83.

Montalvao, J., Almeida, C. A. S., and Freire, E. O. (2006). Equalization of keystroke timing histograms for improved identification performance. In *2006 International telecommunications symposium*, pages 560–565. IEEE.

owasp.org (2020). Credential stuffing. https://owasp.org/ www-community/ attacks/Credential˙stuffing. Accessed: 2020-04-03.

Pisani, P. H. and Lorena, A. C. (2013). A systematic review on keystroke dynamics. *Journal of the Brazilian Computer Society*, 19(4):573–587.

Revett, K., De Magalhães, S. T., and Santos, H. M. (2006). Enhancing login security through the use of keystroke input dynamics. In *International Conference on Biometrics*, pages 661–667. Springer.

Rybnik, M., Panasiuk, P., and Saeed, K. (2009). User authentication with keystroke dynamics using fixed text. In *2009 International Conference on Biometrics and Kansei Engineering*, pages 70–75. IEEE.

Sitová, Z., Šeděnka, J., Yang, Q., Peng, G., Zhou, G., Gasti, P., and Balagani, K. S. (2015). HMOG: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security*, 11(5):877–892.

Teh, P., Teoh, A., and Yue, S. (2013). A survey of keystroke dynamics biometrics. *The Scientific World Journal*.