SN Computer Science (2022) 3: 360
https://doi.org/10.1007/s42979-022-01245-3

ORIGINAL RESEARCH

Securing Account Recovery Mechanism on Desktop Computers and Mobile Phones with Keystroke Dynamics

Ahmed Anu Wahab[1] · **Daqing Hou**[1] · **Stephanie Schuckers**[1] · **Abbie Barbir**[2]

Received: 1 October 2021 / Accepted: 10 June 2022 / Published online: 5 July 2022

Abstract

Account recovery has become a prevalent feature across mobile and web applications that circumvents the regular username/password-based user authentication process, and thus is known to be less secure and fraught with attacks. For example, to trigger the account recovery process, an email or one-time password (OTP) is sent to the user's registration email and/or phone. This assumes that only the genuine user has access to the email/phone which is not always the case. To further improve the security of the account recovery mechanism, beyond validating the information and other credentials typed by the user, we propose a recovery method with the use of keystrokes dynamics. We evaluated performances using two new keystroke datasets—the first contains over 500,000 keystrokes collected on a desktop computer from 44 participants, while the second 327,000 keystrokes on a touchscreen mobile phone from 39 participants. Both datasets require the participants to fill out an account recovery form of multiple fields. For each dataset, we evaluated the performance of five scoring algorithms on individual fields, feature-level fusion and weighted-score fusion. We also applied one-class classification, a machine learning approach and compared results. For the desktop dataset, we achieved the best equal error rate (EER) of 5.47% for individual fields, 0% for feature-level fusion of five fields, and 0% for weighted-score fusion of seven fields. For the touch-mobile dataset, we achieved the best EER of 10.25% for individual fields, 4.97% for feature-level fusion of four fields and 2.26% for weighted-score fusion of seven fields. Our results show that the application of keystroke dynamics is highly promising to further secure the account recovery mechanism on both desktop and mobile platforms.

Keywords Account Recovery · Forgot Password/Username · Keystroke Dynamics · Free-text · Fixed-text · Behavioral Biometrics

Introduction

The username and password not only have been the dominant means of verifying a user's digital identity over the years [1], 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 [2]. In 2019, a collection of 2.7 billion identity records, consisting of 774 million unique email addresses and 21 million unique passwords, was posted on the web for sale [3]. Because of the difficulty in remembering passwords, many users have been known to use a single password across multiple websites and apps, making it even easier for impostors to take over their accounts. To increase security, a common practice has been adopted by many sites and apps, which require users to regularly change their passwords and to use long unique passwords, for example, a combination of

This article is part of the topical collection "Information Systems Security and Privacy" guest edited by Steven Furnell and Paolo Mori.

✉ Ahmed Anu Wahab
wahabaa@clarkson.edu

Daqing Hou
dhou@clarkson.edu

Stephanie Schuckers
sschucke@clarkson.edu

Abbie Barbir
BarbirA@aetna.com

[1] Electrical and Computer Engineering, Clarkson University, Clarkson Avenue, Potsdam 13699, NY, USA

[2] Mobile Security Group, CVS Health, Aetna, Woonsocket, USA

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 and on mobile applications. 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 and apps with low security requirements, to increase the level of security, many sites and apps also require the user to perform additional verification, such as answering security questions or providing personal credentials (Fig. 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 possibility that any impostor with access to a user's recovery email (which can be taken over by attacks such as credential stuffing [5]) can easily trigger an account recovery session and take over the user's account. Given that account recovery is ubiquitous across the web and mobile applications, and widely 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 [6–8] but does not require additional hardware. Our research goal is to further strengthen the security of the account recovery mechanism using keystroke dynamics. Moreover, 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.

In this paper, we focus on sites and apps that have implemented additional verification during account recovery by requesting more information, including the email address, from the user. Using an account recovery form with multiple fields, we have created two new datasets—the first is a dataset with over 500,000 keystrokes collected on a desktop computer from 44 students and university staff, while the second 327,000 keystrokes on a touchscreen mobile phone from 39 students and university staff.

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 return low scores when samples come from the same user and high scores when samples come from different users. We also applied one-class classification (OCC), a machine learning approach, on the touchmobile dataset. The OCC is an outlier or anomaly detection algorithm that tries to identify examples of a specific class amongst all examples, by primarily learning from a training set containing only the examples of that class. Lastly,

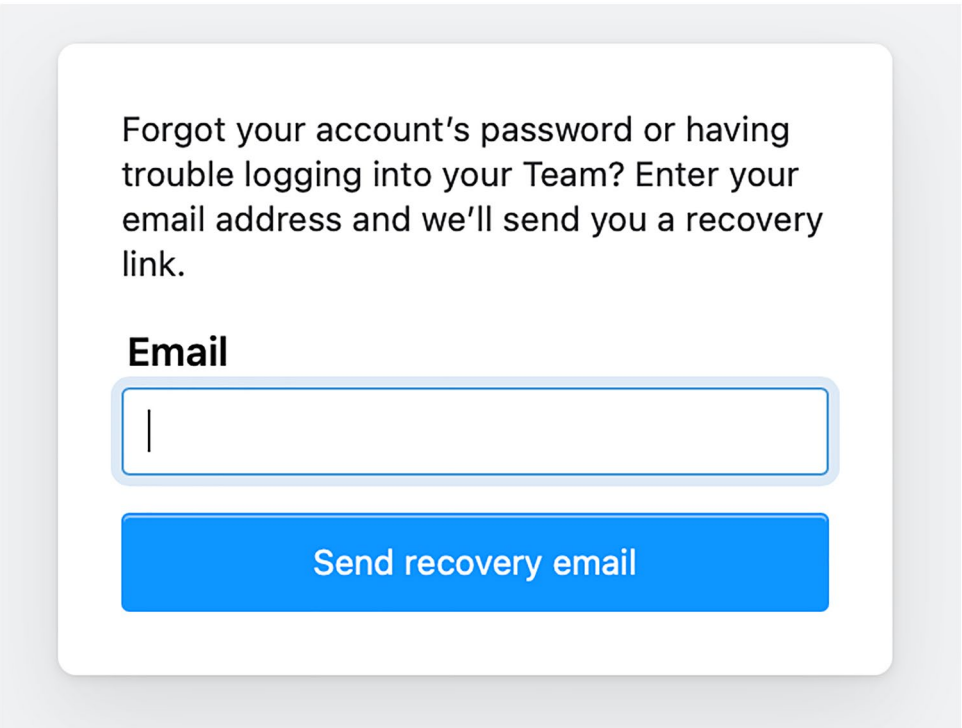

Fig. 1 Stackoverflow password recovery requires a user's verified email address to which the password reset link will be sent

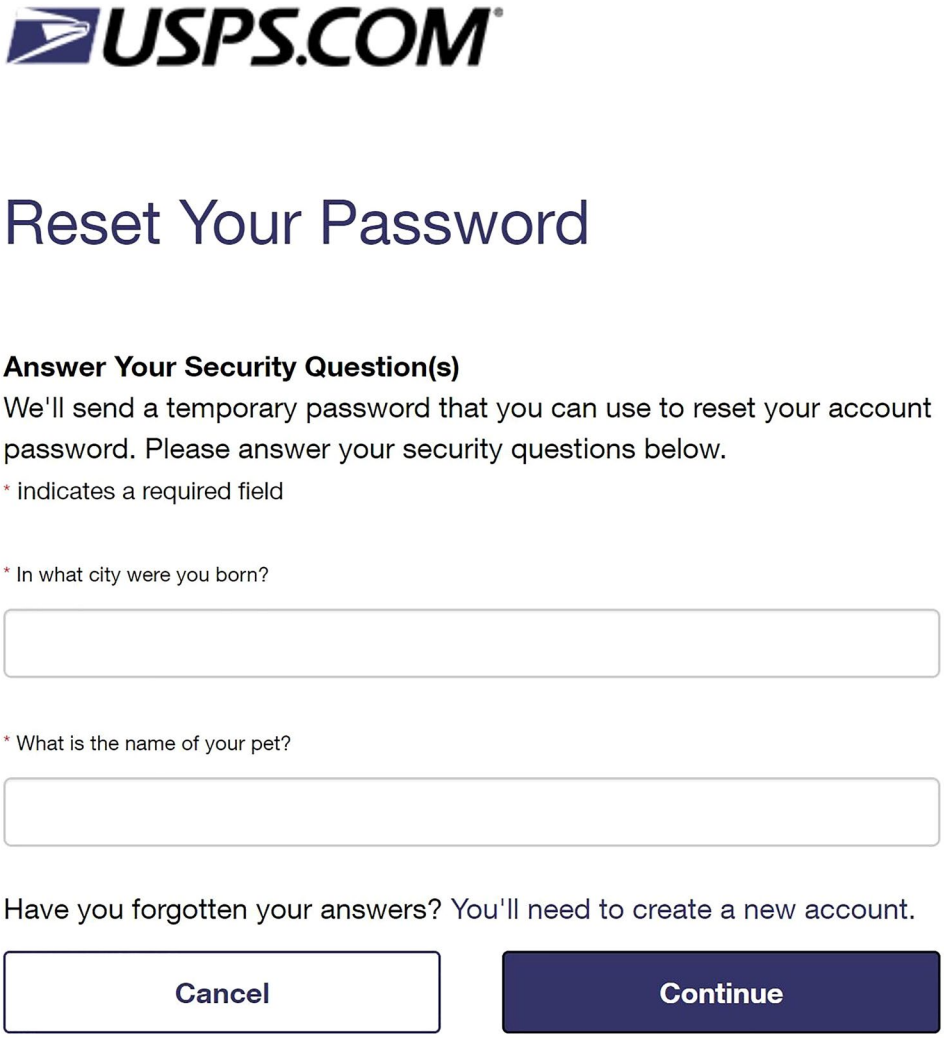

Fig. 2 As additional protection, United State Postal Service (USPS) account recovery also requires a user to answer security questions [4]

we experimented on the minimum number of enrollment samples required to build a user's profile using the desktop dataset.

Using the desktop dataset, we achieve the best EER of 5.47% when using individual fields, and 0% for both feature-level fusion and weighted-score fusion. For the touch-mobile dataset, we achieved the best EER of 10.25% for individual fields, 4.97% for feature-level fusion of four fields and 2.26% for weighted-score fusion of seven fields.

As shown in Fig. 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 datasets is used individually for authentication, this work can be considered as short length, fixed-text keystroke dynamics; but when fields in the datasets 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.

This article significantly improves a preliminary version presented in [4] in the following ways:

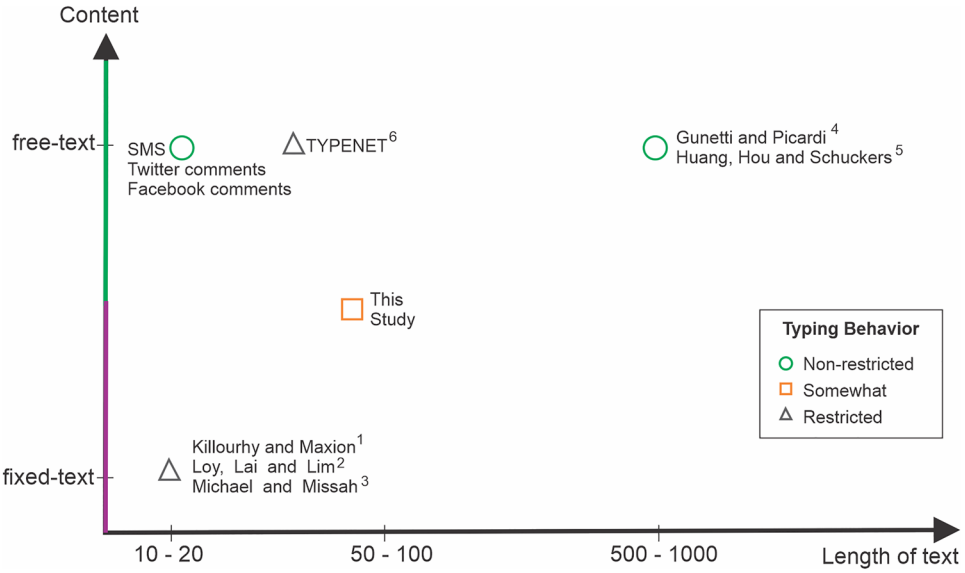

Fig. 3 Characterizing keystroke dynamics based on three traits: Length of text (long or short), typing behaviour (restricted or unrestricted) and typed content (fixed or free). Our study is between fixed-text and free-text in a laboratory setting (somewhat restricted). [1] [9], [2] [10], [3] [11], [4] [12], [5] [13] and [6] [14]

1. We create a new dataset with 39 participants who contributed data by filling an account recovery form on a touchscreen mobile device. This touch-mobile dataset is similar, with the same type and number of fields, as the desktop data presented in [4].
2. We analyse the touch-mobile data to investigate if keystrokes collected from touchscreen mobile devices is sufficient to strengthen the account recovery mechanism, or a fusion with additional behavioural modality, such as touch dynamics, is required.
3. We evaluate performances using the touch-mobile dataset on individual fields and combination of fields through two major fusion techniques—feature level fusion and weighted-score fusion.
4. We further evaluate performances on the touch-mobile data through a machine learning approach known as OCC, an outlier detection algorithm.
5. We compare the results obtained from the touch-mobile data analysis with those in the preliminary work.

The remaining of this paper is organized as follows. Section "Related Work" presents related work in both fixed-text and free-text keystroke dynamics. Section "Methods" describes our methodology: the two datasets, feature extraction, algorithms and implementation procedure. Results and findings are presented in Sect. "Results". Lastly, Sect. "Conclusions" concludes the paper.

Related Work

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. [15] 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 [16–18].

Gunetti and Picardi [12] 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. [13] 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 500–1000 digraph instances have been used in free-text literature (Fig. 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 [12] and Huang et al. [13].

Keystroke dynamic features are extracted using the timing information of keys pressed, which includes latency between consecutive keys and dwell/hold time of a single key. As shown in Fig. 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

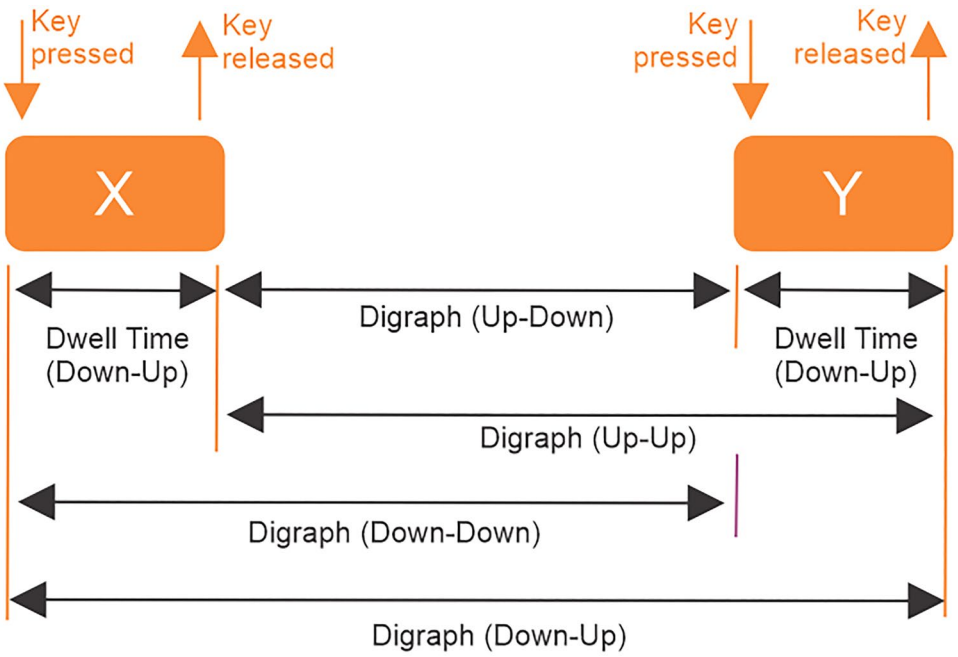

Fig. 4 Keystroke dynamics features (dwell/hold time and digraph latency defined in terms of key press/release events)

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 [8, 19–22] and free-text [12, 23], 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 [9], Loy, Lai and Lim [10], and Michael and Missah [11]. Killourhy and Maxion have a dataset of 20,400 samples, collected from 51 subjects on a desktop computer 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, such 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 [24] create a dataset with samples collected from 83 users (Table 1), a total of 5185 genuine samples (pair of chosen username and password typed by its owner), 5754 impostor samples (pair of username and password typed by a user different of its owner), and 5439 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.

Table 1 Password datasets for keystroke dynamics [4]

Dataset	#Users	#Samples	User Specific Password?
Killourhy and Maxion [9]	51	20,400	No
Giot, El-Abed and Rosenberger [24]	83	5,185+ 5,754/5,439	Yes
BioChaves [26]	47	1,400	No
Allen [27]	104	2,736	No
Keystroke100 [10]	100	1,000	No
GREYC-NISLAB [28]	110	2,201	No

Nader, Zarina and AbedElkarim in [25] proposed the use of interface preferences authentication (UIPA) for strengthening the security of account recovery mechanism. The method was proposed for online systems that offer user interface (UI) design option where users can choose a preferred design based on personal characteristics. The authors reported a false positive rate and false negative rate of 0.416% and 0%, respectively, from 83 participants. This approach is knowledge-based, which requires additional

actions from the users to re-specify their chosen design preferences by filling out a form. If this approach is used in practice, the user may find it difficult to remember their design preferences for multiple websites. As a result, the method is not expected to be widely adopted.

Methods

The account recovery mechanisms implemented on many public and business websites and apps 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 Stackoverflow website requires just a single email address (Fig. 1), an online banking platform, which is more security-sensitive, would request multiple fields of information for added security (Fig. 5). This leaves us with the following research questions:

1. How many is enough? That is, how many fields of information is needed to be collected from users to achieve the desired security for account recovery using keystroke dynamics?
2. What level of information is sufficient to improve security for account recovery?
3. Are all fields of information the same? That is, do information like email, phone number or address contribute equally to having low intra-class but high inter-class variations?

For improved security and for the purpose of finding answers to the above questions, we have collected multiple fields of information from users during our data collection. In biometric authentication, the conflict between security and convenience must be resolved. In other words, the strength of one should not weaken the other. However, the public acceptance of our approach (implementation on a live website or mobile app) and how it affects convenience is out of scope of this paper and will be considered in future study.

Account-Recovery Keystroke Dataset

Data were collected on a desktop computer and a touchscreen mobile device from 44 and 39 participants respectively who are university students and staff.

Desktop Dataset

This dataset was created with a total of over 500,000 keystrokes. The data was collected from 44 university students and staff using a data collection web app with a physical QWERTY keyboard (Fig. 6). Each user visits us twice and data were collected in a laboratory setting. 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

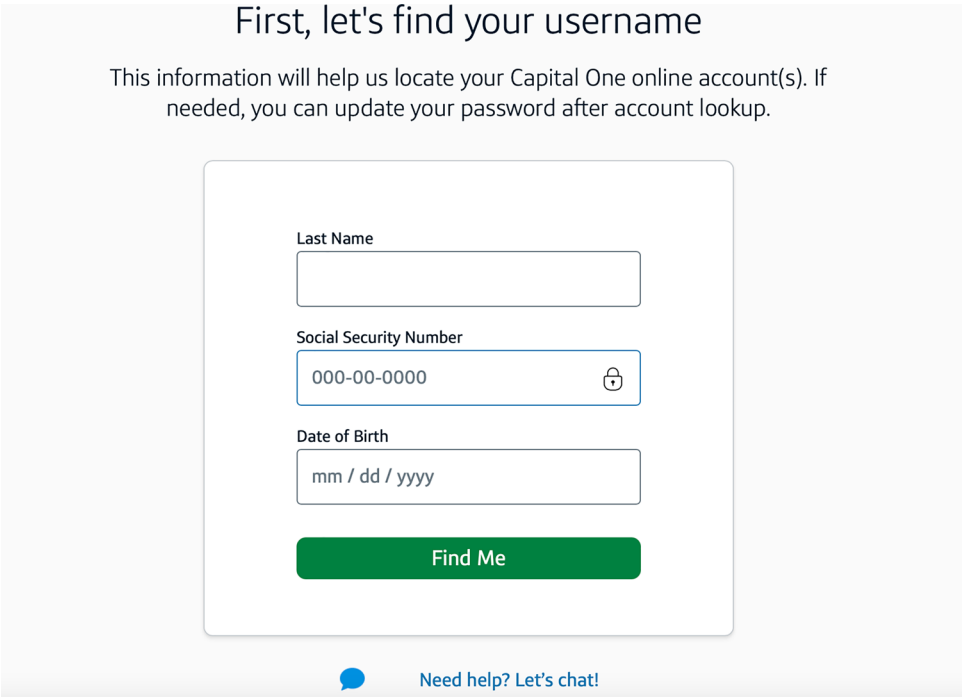

Fig. 5 Capital One forget password session requires a user to enter multiple fields of information

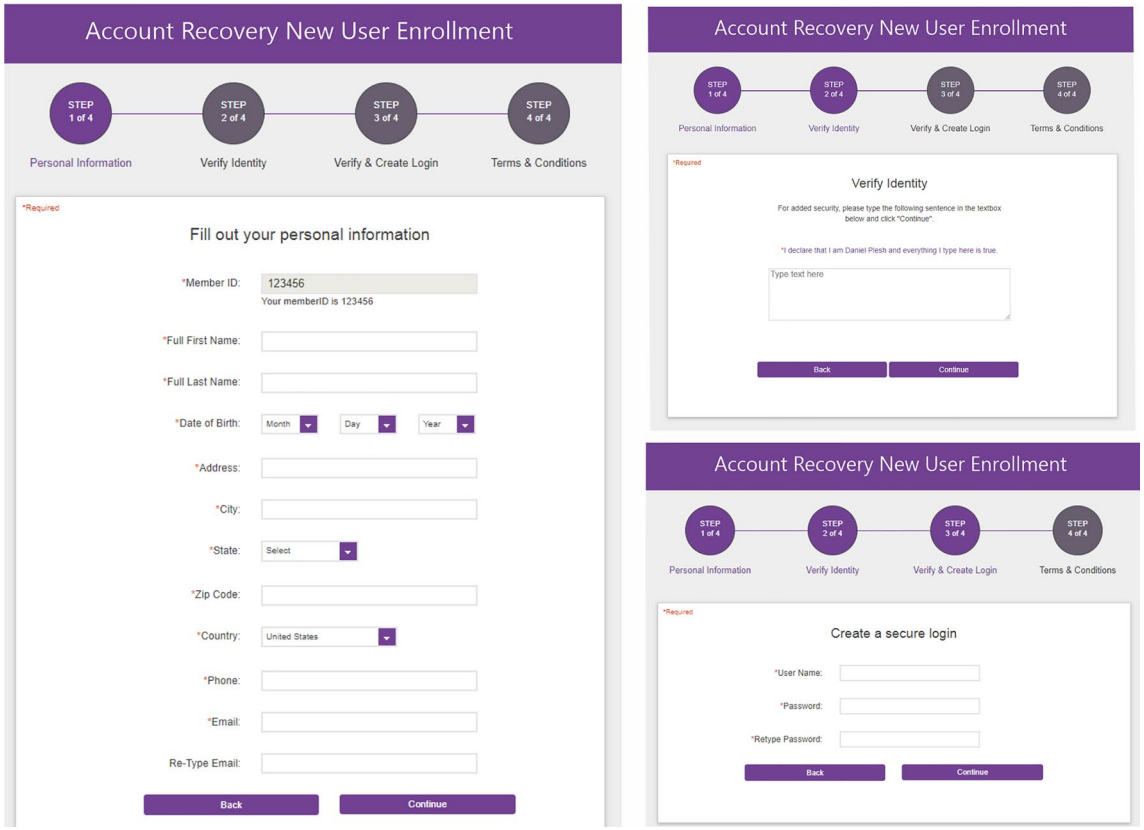

Fig. 6 Account Recovery Desktop dataset: User interfaces of the data collection web app [4]

is used as the user's genuine keystrokes. The same user is given five other users' information to attack with, each twice, which serves as impostor keystrokes. The impostor is considered an informed attacker because he/she has the information and credentials of the true user. As a result, our new dataset contains data for when users attack each other. Fig. 7 depicts such an example where user (ID: W0037-81456) attacks another user W0092-17843. All participants were given some information on the required activities for each visit and were compensated.

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 5 more times, but only 16 of the 28 users have been attacked (Table 2).

Touch-Mobile Dataset

In present day, touchscreen mobile phones make up a larger share in the mobile market and almost all activities (such as banking, online transactions and purchases etc) that are carried out on the web can also be done using touchscreen mobile phones. Therefore, similar to the desktop dataset, we created another Account-Recovery dataset with a total of 327,000 keystrokes, collected from 39 university students and staff on a touchscreen mobile device running on android OS. We developed an application for the purpose of the data collection using the Android Studio IDE (Fig. 8). Participants are required to maintain a sitting position but not restricted to sitting upright for the study. They are also not restricted on how to position the device (on the desk or in their hands) while typing. Each user visited twice and data were collected in a laboratory setting. In the first visit, each user completes an enrollment form using the android app ten times, which is used to build the user's profile. In the second visit, each user completes the same form five times, which is used as the user's genuine samples. Thereafter, the user is given the information and credentials of two other users to attack with, each twice, which is used as impostor samples. The impostor is considered an informed attacker because he/she is given the information and credentials of the true user. The visits are 1 or 2 weeks apart, although few users had both visits 2 or 3 days apart. Towards better data quality and learning from the desktop dataset, we prevented participants from using the copy and paste features, and ensured that user's subsequent session data matches the previous. For example, the user information typed while filling the enrollment form for the second time matches the

	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

Fig. 7 Account Recovery Desktop dataset: User W0037-81456 attacks the profile of user W0092-17843 [4]

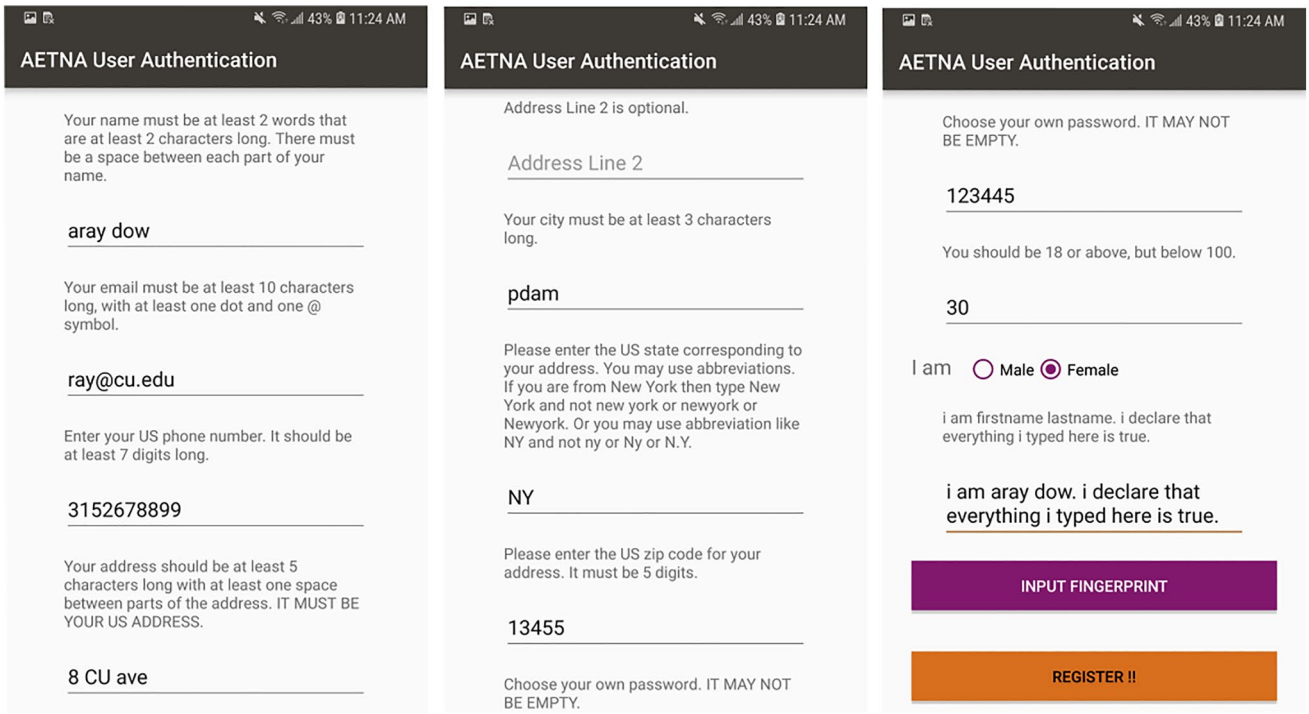

Fig. 8 Account Recovery Touch-Mobile dataset: User interfaces of the data collection mobile app

information provided in the first. All participants were given some information on the required activities for each visit and were compensated. Overall, 31 of 39 participants contributed data on both visits and 36 users were attacked as shown in Table 2.

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 Figs. 6 and 8). The desktop and touch-mobile datasets hold the record of key-down and

Table 2 Number of keys contributed per user after data pre-processing

Dataset		Avg / Min / Max keys per User	#User
Desktop	Profile	2048 / 1282 / 3510	42
	Genuine	1210 / 614 / 3219	28
	Impostor	2351 / 88 / 7615	16
Touch-Mobile	Profile	1996 / 1136 / 2761	39
	Genuine	973 / 537 / 1311	31
	Impostor	1422 / 421 / 2197	36

Table 3 Number of keys contributed per field after data pre-processing

Fields	Desktop Avg / Min / Max keys per Field	Touch-Mobile Avg / Min / Max keys per Field
Full name	13 / 4 / 20	17 / 12 / 30
Address	17 / 8 / 38	27 / 17 / 54
City	9 / 5 / 17	10 / 7 /21
Zip	6 / 5 / 10	6 / 5 / 9
Phone	12 / 10 / 26	10 / 7 / 17
Email	21 / 15 / 37	23 / 18/ 45
Declare Text	68 / 53 / 135	72 / 68 / 97

key-up timing information of every key pressed and released, and all participants are allowed to make and correct typing errors while contributing data.

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. Tables 2 and 3 show a summary of keys contributed per user and per field, respectively, after data cleaning and pre-processing.

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 [24].

Feature Extraction

We extracted features from the keystroke timestamp recorded in the raw Account-Recovery data. Of the many types of feature extraction, below are the commonly used feature extraction.

- Dwell Time (DT)
- Flight Time: Up-Down (UD)
- Digraph: Down-Down (DD)
- Digraph: Up-Up (UU)

In this paper, we have used the Down-Down (DD) digraph feature for all the statistical scoring algorithms, which can be easily obtained from the time difference between successive characters in the dataset. Take for instance the word "password", the DD feature for digraph 'pa' is the time difference between the timestamp recorded when 'p' and 'a' is pressed. Therefore also, the DD feature for digraph 'as' is the time difference between the timestamp recorded when 'a' and 's' are pressed. The same goes for digraph 'ss', 'sw', 'wo', 'or' and 'rd'. In general, for every N character word, there should be N-1 digraphs in that word (without considering digraph repetitions). The word 'password' has 8 characters and 7 digraphs.

Scoring Algorithms

We have implemented five state-of-the-art scoring algorithms from both fixed text and free text keystroke dynamics [9, 12, 23, 29]: Euclidean Distance, Manhattan Distance, Scaled Manhattan Distance, Mahalanobis Distance, and the 'A and R' Measures of Gunetti and Picardi [12].

Figure 9 illustrates how the scoring algorithms work. Note that in the sample text 'mississippi', the digraph 'is', 'ss' and 'si' are repeated twice, but 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 Fig. 9. 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.

Euclidean Distance

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

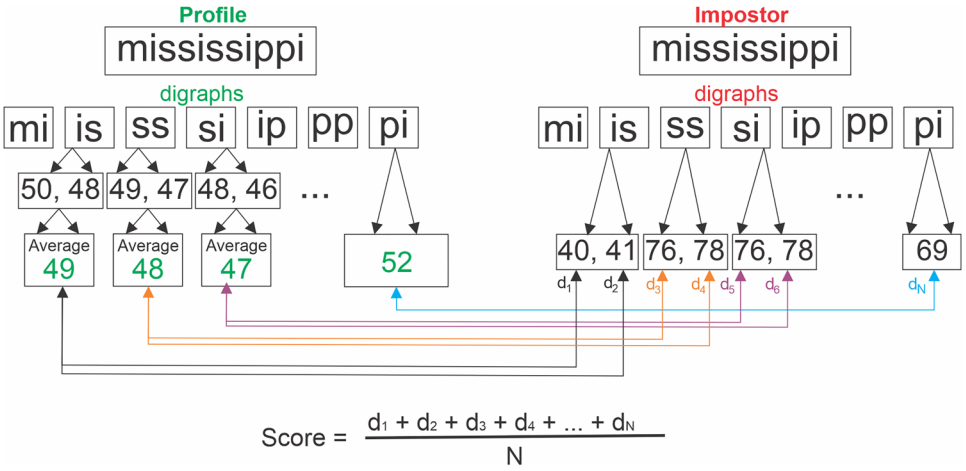

Fig. 9 Scoring procedure for sample text 'mississippi' where d_i represents the timing difference between the *ith* digraph in the test sample and the profile [4]

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

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} is the mean of the i^{th} graph in the profile.

Manhattan Distance

The scaled Manhattan and Manhattan distance metrics were used by Kilhourhy and Maxion for fixed-text keystroke dynamics [9]. The scaled Manhattan distance is calculated as follows:

$$D = \sum_{i=1}^{N} \frac{\|\mu_{g_i} - x_i\|}{\sigma_{g_i}}$$

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 [9]. The Manhattan and scaled Manhattan distances are identical, except the Manhattan distance is not divided by the standard deviation [30].

Mahalanobis Distance

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

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

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} and σ_{g_i} are the mean and standard deviation of the i^{th} graph in the profile [9] and [31].

Gunetti and Picardi's Metric

Gunetti and Picardi's free-text algorithm [12] combines typing speed (A-measure) and the degree of disorder (R-measure) to measure similarity [23]. 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:

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

where t is a constant for determining n-graph similarity. For example, let $G_{S1,L1}$ and $G_{S2,L2}$ be the same n-graph occurring in typing samples S1 and S2, with latencies L1 and L2, respectively. We say that $G_{S1,L1}$ and $G_{S2,L2}$ are similar if and only if $1 \leq \max(L1,L2)/\min(L1,L2) \leq t$. 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′.

Experiments

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 weighted-score levels. The result of each experiment is presented and discussed in Sect. "Results".

The desktop dataset 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 with the desktop dataset and recorded the K that produces the lowest EER.

On the other hand, for the touch-mobile dataset, we enforced the quality control mechanism for each field when the data was collected. This ensures that the Fullname has 100% match; while Email, Address, Zip, City and Phone have at least a 95% content match between data contributed in subsequent sessions and the first. Finally, the Declare field requires 85% content matching, which means participants have an allowable uncorrected typo of 15%. Otherwise, the participant is required to repeat the session.

Individual Fields

Do all fields of information contribute equally in telling users apart? Results from the individual fields experiment answer this question. Here, we treat each field individually and compare only the profile and test samples of the same field. The EER for each field is then recorded.

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 best performance (the lowest EER). Specifically, we merge all the keystrokes from multiple fields and apply the scoring algorithms. 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).

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 minimum weight is 0.05 with an increment of 0.05 after every search iteration. The grid-search approach is known to perform well for finding optimum weights in behavioral biometrics [32].

One-Class Classification (OCC)

The one-class classifier is a machine learning approach used for outlier detection. Unlike multi-class classifiers, the OCC tries to identify objects of a specific class amongst all examples, by primarily learning from a training set containing only the examples of that class. In the training phase, OCC is fit on data that only has examples from the normal (inlier) class. The trained model is then used to classify new examples as either normal (inliers) or anomalies (outliers). The One-class SVM algorithm is used for the one-class classification. One-class SVM is a variation of the SVM that can be used in an unsupervised setting for anomaly detection. There are three (3) major parameter tuning in One-class SVM which are kernel, gamma and nu. The kernel specifies the type of kernel to be used in the algorithm; gamma is the coefficient of the kernel; and nu, which is in the interval 0 and 1, is the upper bound on the fraction of training errors and a lower bound of the fraction of support vectors. We used the Radial Basis Function (RBF) kernel, and set the value of gamma to $1/(no\_of\_features * X.var())$, where X.var() is the variance of the training data. We used the grid-search approach, ranging from 1 to 50% with an increment of 0.5%, to find the best value for nu that produces the best performance.

The touch-mobile dataset was preprocessed and we converted the key in each keystroke to its respective keycode. The keycodes are in accordance to the ASCII codes with integers in the range of 0 and 255. The keycodes are then normalized by dividing each keycode by 255, which forces the keycodes to be between 0 and 1. Normalization is a common approach in machine learning which allows the model to more quickly learn the optimal parameters for each input. Thereafter, five features were extracted from the data which are: the Up-Down (UD) digraph which is the latency between two consecutive keys; the monographs (m1 and m2) of the two keys, which are the elapsed time between the press and release of each single key; and the normalized keycodes (keycode1 and keycode2) of the two keys.

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.

Results

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

Consistent with the state-of-the-art in fixed-text keystroke dynamics [9], as shown in Figure 10, 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 statistical approach experiments in this paper uses the Scaled Manhattan Distance.

Out of the 7 fields in our account recovery form, there are 21 combinations for Duet (2 fields), 35 combinations for Trio (3 fields), 35 combinations for Quartet (4 fields), 21 combinations for Quintet (5 fields), 7 combinations of Sextet (6 fields) and 1 for Septet (7 fields). We have recorded only the best performance for each of the above field combinations.

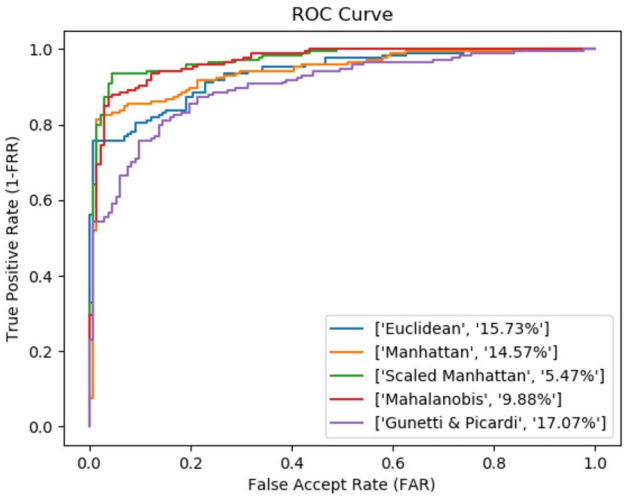

Fig. 10 Receiver Operating Characteristics (ROC) curve for all five algorithms based on the *Declare* field, with Scaled Manhattan Distance being the best (EER of 5.47%) [4]

Results for Desktop Dataset

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 average shared digraphs of 51, 20, and 16, respectively. The 'Zip' field has the lowest accuracy with EER of 22.8%, with only an average of 4 shared 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 and address, are more likely to reveal a user's typing pattern.

Feature-Level Fusion

Consistent with the observed impact of the length of text on accuracy, an overall trend in Table 6 is that performance

Table 5 Authentication based on individual fields for the desktop dataset using the Scaled Manhattan distance algorithm

Field	#Avg shared digraph	K	EER (%)
Zip	4	90%	22.80
City	7	70%	20.36
Phone	8	70%	18.02
Fullname	12	70%	14.16
Address	16	70%	**10.81**
Email	20	70%	**8.10**
Declare	51	70%	**5.47**

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

Field	Euclidean Distance (%)	Manha Distance (%)	Scaled Manhattan Distance (%)	Mahalanobis Distance (%)	Gunetti & Picardi (%)
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

The best performances are shown in Bold

Table 6 Feature level fusion of multiple fields for the desktop dataset using the Scaled Manhattan distance algorithm

Field	#Avg shared digraph	K	EER (%)
DUET			
Email+Fullname	29	90%	4.88
TRIO			
Declare+Email +Address	78	70%	3.13
QUARTET			
Declare+Email+ Address+Fullname	82	70%	2.36
QUINTET			
Declare+Email +Address+ Fullname+City	90	90%	**0.00**
SEXTET			
Declare+Email +Address+Fullname+ City+Zip	95	90%	0.00
SEPTET			
Declare+Email +Address+Fullname+ City+Zip+Phone	102	70%	2.18

The best performance is shown in Bold

Table 7 Weighted-score fusion of multiple fields for the desktop dataset using the Scaled Manhattan distance algorithm, where w is the weight

Field	K	EER (%)
PAIR		
Email(w=0.75)+Declare(w=0.25)	70%	4.3
TRIO		
Email(w=0.55)+Declare(w=0.25) +Fullname(w=0.2)	70%	2.7
QUARTET		
Email(w=0.45)+Declare(w=0.25)+ Fullname(w=0.15)+Address(w=0.15)	70%	2.27
QUINTET		
Email(w=0.45)+Declare(w=0.25) +Fullname(w=0.1)+ Address(w=0.15)+Zip(w=0.05)	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)	70%	1.4
SEPTET		
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=0.1)	80%	**0.00**

The best performance is shown in Bold

improves as the number of shared digraph increases. 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 individual 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.

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 an EER of 0%. 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. Overall, we believe the weighted-score fusion is a better choice for our application because it uses more data and produces better performances when a more strict content matching is applied (*K* is 80% for the combination of seven fields - Septet).

In general, these results (feature-level fusion and weighted-score fusion) outperform the state-of-the-art in both fixed-text and free-text keystroke dynamics. Specifically, the best-performance EERs recorded in fixed-text papers like Killourhy and Maxion [9], and Giot, EL-Abed and Rosenberger [24] are 9.6% and 8.87%, 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 and weighted-score fusion, which outperform the results recorded in free-text papers like Gunetti and Picardi [12] and Huang et al. [13, 23].

Results for Touch-Mobile Dataset

Individual Fields

The results of the individual fields for touch-mobile dataset using the Scaled Manhattan Distance are shown in Table 8. 'Declare', 'Address' and 'Fullname' are the three best performing fields with an EER of 12.36%, 13.1%, and 15.29%, and an average of shared digraphs of 43, 16, and 12, respectively. Similar to the desktop dataset, the field 'Zip' has the

Table 8 Authentication based on individual fields for the touch-mobile dataset using the Scaled Manhattan distance algorithm

Field	#Avg shared digraph	EER (%)
Zip	3	32.17
City	6	20.86
Phone	6	25.13
Fullname	12	**15.29**
Address	16	**13.1**
Email	14	19.42
Declare	43	**12.26**

The best performances are shown in Bold

lowest EER of 32.17%, and only an average of 3 shared digraphs. This confirms our hypothesis that short texts with a small number of shared digraphs produce worse performance. In other words, the longer the text, the better the performance.

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. Familiar contents such as address, email and fullname, are more likely to reveal a user's unique typing pattern.

Feature-Level Fusion

Similar to the feature-level fusion results for the desktop data on the impact of the length of text on accuracy, the trend in Table 9 is that performance improves (EER decreases) as the average number of shared digraphs increases, except for the combinations of five to seven fields. The feature-level fusion performance becomes worse as more data is added. Although further investigation is required to know the exact cause of this observation, we hypothesize that, for the feature-level fusion, performance increases when fields with good individual performances are combined, but worsens when fields with poor individual performances are added, despite the increase in the average number of shared digraphs.

One would have expected that the best Trio combination will be Fullname+Declare+Address since these are the best performing individual fields, and not Email+Declare+Address as seen in the table. However, it is important to point out that participants' full names are also included in the Declare field. Therefore, combining Fullname and Declare using Feature-level fusion gives no new information, although Fullname had a better individual performance than Email. The best performance recorded is 4.97% EER at the combination of four fields (Quartet) with an average shared digraph of 60. These results provide insights on the most important information to be requested from mobile users during account recovery.

Table 9 Feature level fusion of multiple fields for the touch-mobile dataset using the Scaled Manhattan distance algorithm

Field	#Avg shared digraph	EER (%)
DUET		
Declare+Address	56	7.47
TRIO		
Email+Declare+Address	63	5.74
QUARTET		
Email+Declare+ Address+Zip	66	**4.97**
QUINTET		
Declare+Fullname+ Address+City+Phone	68	4.99
SEXTET		
Email+Declare +Fullname+Address+ City+Zip	72	5.07
SEPTET		
Email+Declare +Fullname+Address+ City+Zip+Phone	77	5.15

The best performance is shown in Bold

Weighted-Score Fusion

Table 10 shows the results for the weighted-score fusion experiments. As the field combination increases from Duet (two fields) to Septet (seven fields), performance improves accordingly. The weighted-score fusion follows the hypothesis, with no exception, that more data (longer text) results in better performance. The overall best performance of 2.64% EER is recorded when seven fields are combined (Septet). Overall, the weighted-score fusion is to be preferred because the results are consistent in that including additional data gives better performance. The downside to the weighted-score fusion in this experiment is that the fields combination with the best performance from Duet to Septet do not follow the pattern of being inclusive. For instance, the best Trio (Email+Declare+Fullname) does not include all the fields from the best Duet (Declare+Address).

In general, as seen from our results, keystroke dynamics performances for desktop data are better than those from touch-mobile data [14]. This can be explained by the few discrepancies between the two platforms. First is the keyboard type: one uses physical keyboards while the other virtual keyboards; second is the keyboard size; third is typing position: most people type on the desktop while seated and

Table 10 Weighted-score fusion of multiple fields for the touch-mobile dataset using the Scaled Manhattan distance algorithm, where w is the weight

Field	EER (%)
DUET	
Declare(w=0.75)+Address(w=0.25)	5.28
TRIO	
Email(w=0.45)+Declare(w=0.25) +Fullname(w=0.3)	5.14
QUARTET	
Fullname(w=0.4)+Address(w=0.35)+ City(w=0.15)+Phone(w=0.1)	3.65
QUINTET	
Email(w=0.1)+Declare(w=0.2) +Fullname(w=0.4)+ Address(w=0.15)+City(w=0.15)	3.44
SEXTET	
Email(w=0.2)+Declare(w=0.3) +Fullname(w=0.2)+Address(w=0.15)+ City(w=0.05)+Zip(w=0.1)	2.81
SEPTET	
Email(w=0.2)+Declare(w=0.25) +Fullname(w=0.2)+Address(w=0.2)+ City(w=0.05)+Zip(w=0.05)+Phone(w=0.05)	**2.64**

The best performance is shown in Bold

the keyboard on a flat surface (desk) but typing on mobile devices can be done in any position (standing, sitting, resting, etc); fourth is typing hands and fingers: two hands are majorly used on desktop computers but one hand or even one finger can be used effectively for typing on mobile devices. However, keystroke dynamics performance for touch-mobile may be improved when fused with other data collected from the mobile device through its available sensors such as gyroscope and accelerometer.

One-Class Classification for Touch-Mobile Dataset

Individual Fields

The one-class classification results for individual fields are shown in Table 11. Unlike the results from the statistical model, the Phone field had the best performance with an EER of 10.25% , followed by the Email and Declare fields with EER of 14.83% and 20.40%, respectively.

Feature-Level Fusion

Table 12 shows the one-class classification results with feature-level fusion. The best performance recorded is in the combination of two fields, namely Phone+City, with an

Table 11 One-class classification authentication results based on individual fields for the touch-mobile dataset

Field	Parameter nu (%)	EER (%)
Zip	25.5	22.27
City	16	27.76
Phone	13.5	**10.25**
Fullname	12	21.76
Address	6	25.92
Email	6	**14.83**
Declare	1	**20.40**

The best performances are shown in Bold

EER of 8.7%. Note that this is achieved with short length texts like Phone and City, having a combined average keystrokes of 20. This result also outperforms related work in [24] which fused users own username and password and produced an EER of 11.45%. Similar to our previous observation with feature-level fusion for the Scaled Manhattan algorithm, the feature-level fusion performs worse as more data is added.

Weighted-Score Fusion

Recall that for weighted-score fusion, each field scores are weighted before they are combined with other fields; therefore, it is appropriate that the fields maintain their respective

Table 12 Feature-level fusion of multiple fields for the touch-mobile dataset using one-class classification

Field	Parameter nu (%)	EER (%)
DUET		
Phone+City	12.5	**8.7**
TRIO		
Email+Phone+City	8.5	9.59
QUARTET		
Declare+Fullname+ Phone+Zip	23.5	9.33
QUINTET		
Declare+Fullname+ Phone+Zip+Address	13.5	10.03
SEXTET		
Email+Declare +Fullname+Phone+ City+Zip	8	10.6
SEPTET		
Email+Declare +Fullname+Phone+ City+Zip+Address	7.5	10.94

The best performance is shown in Bold

Table 13 Weighted-score fusion of multiple fields for the touch-mobile dataset using one-class classification, where w is the weight

Field	EER (%)
DUET	
Email(w=0.8)+Phone(w=0.2)	9.04
TRIO	
Email(w=0.4)+Declare(w=0.5) +Phone(w=0.1)	8.1
QUARTET	
Email(w=0.5)+Phone(w=0.3)+ City(w=0.1)+Zip(w=0.1)	8.54
QUINTET	
Email(w=0.4)+Declare(w=0.1) +Phone(w=0.3)+ City(w=0.1)+Zip(w=0.1)	**7.47**
SEXTET	
Email(w=0.3)+Fullname(w=0.1) +Declare(w=0.1)+Phone(w=0.2)+ City(w=0.1)+Address(w=0.2)	8.36
SEPTET	
Email(w=0.2)+Fullname(w=0.1) +Declare(w=0.1)+Phone(w=0.3)+ City(w=0.1)+Zip(w=0.1)+Address(w=0.1)	10.22

The best performance is shown in Bold

nu parameter reported in Table 11. That is, 6% for email, 13.5% for phone and so on. Table 13 shows the result of the weighted-score fusion. The best performance of 7.47% EER is reported at the combination of five fields (Quintet), namely Email+Declare+Phone+City+Zip. Unlike the Scaled Manhattan algorithm, the weighted-score fusion for the One-class classification approach does not completely follow the hypothesis that more data (longer text) results in better performance. Exceptions were seen at Quartet, Sextet and Septet.

In general, the Scaled Manhattan algorithm which is a statistical approach performed better than the one-class classification approach. The exact causes of this difference will remain as future work.

Results for Number of Enrollment Samples

Tables 14 and 15 show the results of our experiment on the minimum number of enrollment samples using the feature-level fusion and weighted-score fusion, respectively, for the desktop dataset. The desktop dataset is used for this experiment. This is because our best performances from previous experiments were recorded using the desktop dataset. 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 14, 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

Table 14 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 15 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

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 to accomplish the same level of accuracy.

Conclusions

We propose to secure account recovery on both desktop and mobile platforms with keystroke dynamics. To that end, we evaluated five scoring algorithms on our desktop and touch-mobile account recovery datasets and found Scaled Manhattan Distance to be the best. We also applied one-class classification, a machine learning approach for outlier detection, on the touch-mobile dataset and compared results. For the desktop dataset, we achieve the best EER of 5.47% for individual fields, a global best EER of 0% when five fields are combined using feature-level fusion, and 0% for weighted-score fusion with all seven fields combined. For the touch-mobile dataset, we achieved 10.25% EER for the best individual field from the one-class classification algorithm, a global best EER of 4.97% when four fields are combined with feature-level fusion using the Scaled Manhattan algorithm, and 2.26% for weighted-score fusion with all seven fields combined using the Scaled Manhattan algorithm. Overall, the statistical approach (Scaled Manhattan algorithm) performs the best with the lowest global EER.

In deciding the number of enrollment samples needed to build a user's profile, we found 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 on desktop and mobile platforms. Hence, it is highly promising to apply keystroke dynamics to secure the ubiquitous account recovery mechanism on both desktop and mobile platforms.

In keystroke dynamics, there are possibilities of inconsistent keystrokes which may be caused by cramped muscles, sweaty hands, or change of keyboards [33, 34]. In such cases, keystroke dynamics would reject the users and request them to present another authentication factor. It is important to stress that other MFA (Multi-factor Authentication) factors 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 factors only when the user is rejected, such as in the case of cramped muscles, sweaty hands, or change in keyboards.

Future work includes testing with larger datasets with more samples per user and more users to further validate our techniques, implementing our techniques on a live website and/or mobile app to perform usability testing with real users and survey its public acceptance. Also, monograph and digraph features other than 'DD' can be used for possible

further improvement of performance. Lastly, we will investigate the fusion keystroke dynamics 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.

Acknowledgements 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 desktop data and performed a preliminary performance investigation in 2018. Aratrika Ray collected the mobile dataset.

Author Contributions Not applicable.

Funding Information Not applicable.

Availability of Data and Materials Not applicable.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

Ethics Approval Not applicable.

Consent to Participate Not applicable.

Consent for Publication Not applicable.

Code Availability Not applicable.

References

1. Bonneau J, Herley C, Van Oorschot PC, Stajano F. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In: 2012 IEEE Symposium on Security and Privacy, 2012; 553–567. IEEE
2. Gemalto Inc: 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, 2018;
3. Song V. Mother of All Breaches Exposes 773 Million Emails, 21 Million Passwords. https://gizmodo.com/mother-of-all-breaches-exposes-773-million-emails-21-m-1831833456. Accessed: 2019-09-20
4. Wahab AA, Hou D, Schuckers S, Barbir A. Utilizing keystroke dynamics as additional security measure to protect account recovery mechanism. In: ICISSP, 2021;33–42
5. owasp.org: Credential stuffing. https://owasp.org/ www-community/ attacks/Credential_stuffing. Accessed: 2020-04-03, 2020
6. Rybnik M, Panasiuk P, Saeed K. User authentication with keystroke dynamics using fixed text. In: 2009 International Conference on Biometrics and Kansei Engineering, 2009; 70–75. IEEE
7. Choraś M, Mroczkowski P. Keystroke dynamics for biometrics identification. In: International Conference on Adaptive and Natural Computing Algorithms, 2007;424–431. Springer
8. Revett K, De Magalhães ST, Santos HM. Enhancing login security through the use of keystroke input dynamics. In: International Conference on Biometrics, 2006;661–667. Springer
9. Killourhy KS, Maxion RA. Comparing anomaly-detection algorithms for keystroke dynamics. In: 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, 2009;125–134. IEEE
10. Loy CC, Lai WK, Lim CP. Keystroke patterns classification using the artmap-fd neural network. In: Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007), 2007;1:pp. 61–64. IEEE
11. Michael OB, Missah YM. Utilizing keystroke dynamics as an additional security measure to password security in computer web-based applications-a case study of uew. Int J Comput Appl. 2016;149(5):35–44.
12. Gunetti D, Picardi C. Keystroke analysis of free text. ACM Trans Inform Syst Secur (TISSEC). 2005;8(3):312–47.
13. Huang J, Hou D, Schuckers S, Hou Z. Effect of data size on performance of free-text keystroke authentication. In: IEEE International Conference on Identity, Security and Behavior Analysis (ISBA 2015), 2015;1–7. IEEE
14. Acien A, Morales A, Monaco JV, Vera-Rodriguez R, Fierrez J. Typenet: Deep learning keystroke biometrics. arXiv preprint arXiv:2101.05570,2021;
15. Gaines RS, Lisowski W, Press SJ, Shapiro N. Authentication by keystroke timing: some preliminary results. Rand Corp Santa Monica CA: Technical report; 1980.
16. Banerjee SP, Woodard DL. Biometric authentication and identification using keystroke dynamics: A survey. J Pattern Recognit Res. 2012;7(1):116–39.
17. Teh PS, Teoh ABJ, Yue S. A survey of keystroke dynamics biometrics. The Scientific World Journal; 2013.
18. Alsultan A, Warwick K. Keystroke dynamics authentication: a survey of free-text methods. Int J Comput Sci Issues (IJCSI). 2013;10(4):1.
19. Pisani PH, Lorena AC. A systematic review on keystroke dynamics. J Brazilian Comput Soc. 2013;19(4):573–87.
20. Monrose F, Reiter MK, Wetzel S. Password hardening based on keystroke dynamics. Int J Inform Secur. 2002;1(2):69–83.
21. Bartlow N, Cukic B. Evaluating the reliability of credential hardening through keystroke dynamics. In: 2006 17th International Symposium on Software Reliability Engineering, 2006;117–126. IEEE
22. de Magalhaes ST, Revett K, Santos HM. Password secured sites-stepping forward with keystroke dynamics. In: International Conference on Next Generation Web Services Practices (NWeSP'05), 2005; 6. IEEE
23. Huang J, Hou D, Schuckers S, Law T, Sherwin A. Benchmarking keystroke authentication algorithms. In: 2017 IEEE Workshop on Information Forensics and Security (WIFS), 2017;1–6. IEEE
24. Giot R, El-Abed M, Rosenberger C. Web-based benchmark for keystroke dynamics biometric systems: A statistical analysis. In: 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2012;11–15. IEEE
25. Karim N.A, Shukur Z, AL-banna A.M. Uipa: User authentication method based on user interface preferences for account recovery process. J Inform Secur Appl. 2020;52:102466.
26. Montalva J, Almeida CAS, Freire EO. Equalization of keystroke timing histograms for improved identification performance. In: 2006 International Telecommunications Symposium, 2006; 560–565. IEEE
27. Allen JD. An analysis of pressure-based keystroke dynamics algorithms. PhD thesis, Southern Methodist University, 2010;
28. Idrus SZS, Cherrier E, Rosenberger C, Bours P. Soft biometrics database: A benchmark for keystroke dynamics biometric systems.

In: 2013 International Conference of the BIOSIG Special Interest Group (BIOSIG), 2013;1–8. IEEE
29. Killourhy K, Maxion R. Why did my detector do that?! In: International Workshop on Recent Advances in Intrusion Detection, 2010; 256–276. Springer
30. Black PE. Manhattan distance. Available online at: https://www.nist.gov/dads/HTML/manhattanDistance.html. Last Accessed: 2019-06-15, 2019
31. Mahalanobis PC. On the generalized distance in statistics. National Institute of Science of India, 1936;
32. Sitová Z, Šeděnka J, Yang Q, Peng G, Zhou G, Gasti P, Balagani KS. HMOG: New behavioral biometric features for continuous authentication of smartphone users. IEEE Trans Informat Forensics Secur. 2015;11(5):877–92.
33. Bours P, Ellingsen J. Cross keyboard keystroke dynamics. In: 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), 2018; 1–6. IEEE
34. Wahab AA, Hou D, Banavar M, Schuckers S, Eaton K, Baldwin J, Wright R. Shared multi-keyboard and bilingual datasets to support keystroke dynamics research. In: Proceedings of the Twelveth ACM Conference on Data and Application Security and Privacy, 2022; pp. 236–241