Information Sciences 308 (2015) 72–93

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Keystroke dynamics-based user authentication using long and free text strings from various input devices

Pilsung Kang [a,*], Sungzoon Cho [b]

[a] School of Industrial Management Engineering, Korea University, Seoul, South Korea
[b] Dept. of Industrial Engineering, College of Engineering, Seoul National University, 151-744, 599 Gwanangno, Gwanak-gu, Seoul, South Korea

ARTICLE INFO

Article history:
Received 23 July 2013
Received in revised form 20 March 2014
Accepted 25 August 2014
Available online 18 September 2014

Keywords:
Keystroke dynamics
User authentication
Long and free text strings
Various input devices
Cramér–von Mises criterion
R and *A* measure

ABSTRACT

Keystroke dynamics, which refers to the typing pattern of an individual, has been highlighted as a practical behavioral biometric feature that does not require any additional recognition device for strengthening user authentication or identification. However, research in the area of keystroke dynamics-based user authentication (KDA) has been primarily focused only on the short predefined text, such as identification (ID) and password, typed on a traditional personal computer (PC) keyboard. In this paper, we aim to explore the extendability of KDA by considering long and free text strings from various input devices. Three fundamental questions are raised about the dependence of authentication performance on (1) the type of input device, (2) the length of text strings, and (3) the type of authentication algorithm. Based on the experimental tests, we observe that (1) the usage of a PC keyboard reported the highest authentication accuracy, followed by a soft keyboard and a touch keyboard; (2) the authentication accuracy could be strengthened by increasing the length of either reference or test keystrokes; (3) the $R+A$ and RA measures report the best performance with a PC keyboard, while the Cramér–von Mises criterion reports the best performance with the other input devices for most cases, followed by the Parzen window density estimator.

1. Introduction

Today, we live in a ubiquitous network world, with the ability to connect to network systems (e.g. the World Wide Web (WWW)), irrespective of time and location, with the help of various information technology (IT) devices [19,58]. This increase in connectivity has triggered concerns about the security of network systems in protecting personal information and private data of users [8,27,38]. Contrary to the expectations of users, many network systems are vulnerable, and have a relatively low level of security. A password has been the most common solution for user authentication or identification in network systems [5,65]. Although password-based user authentication systems have many benefits, such as ease of development, maintenance, and operation, as well as cost-effectiveness, they become vulnerable when a third-party acquires the ID and password of a valid user [33,65]. The system cannot prevent an impostor who acquires the log-in information illegally or even unintentionally from accessing the system and acting as if he were the valid user.

In order to overcome the vulnerability of password-based authentication systems, two-stage authentication systems have been developed based on some possessive biometric features such as voice, fingerprint, and iris [10,13,16,39,40,48,51,62].

[*] Corresponding author.
E-mail addresses: pilsung.kang@gmail.com (P. Kang), zoon@snu.ac.kr (S. Cho).

http://dx.doi.org/10.1016/j.ins.2014.08.070
0020-0255/© 2014 Elsevier Inc. All rights reserved.

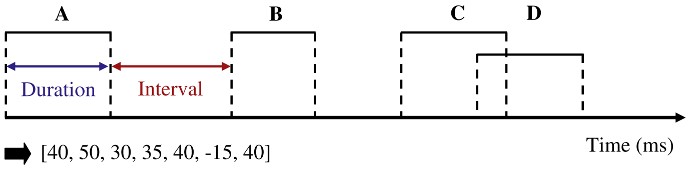


Fig. 1. Keystroke dynamics example of four letter-word “ABCD”.

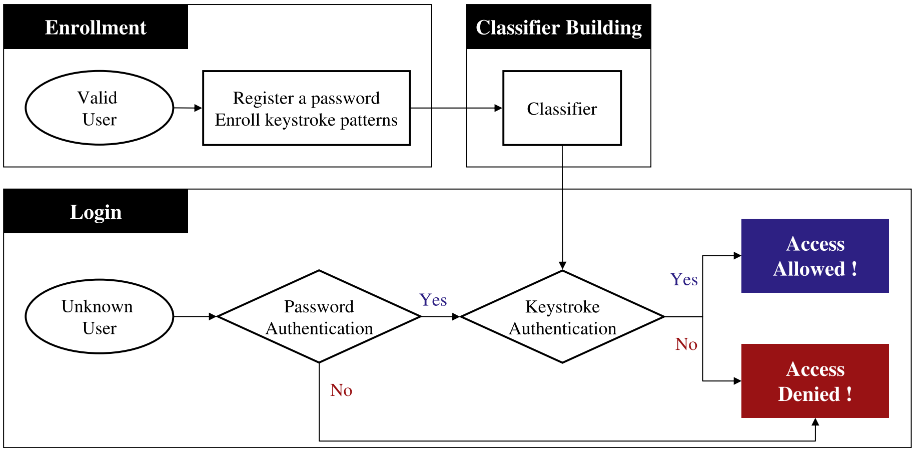


Fig. 2. The general architecture of a keystroke dynamics-based user authentication (KDA) system.

Although these two-stage authentication systems could improve the authentication performance, the adoption rate for such systems is restricted because of the requirement for installing additional recognition device to the current system. In recent times, KDA has been highlighted as an alternative to possessive biometrics-based two-stage user authentication systems. Because it is based on a behavioral biometric feature, i.e., keystroke dynamics, it can be fully implemented in software, which in turn, can be applicable to any systems without installing additional hardware [7,17,36,43–46,64]. Keystroke dynamics is defined as the manner of typing a string of characters as shown in Fig. 1, where the typing action can be transformed into a one-dimensional vector [40, 50, 30, 35, 40, -15, 40]. The keystroke dynamics of one person is significantly different from those of others due to several factors, such as age, primary language, familiarity with the input device, etc. Fig. 2 illustrates the general architecture of a KDA system. During creation of an account, a new user first specifies his ID and password, and then provides a certain number of typing patterns until the system is able to build an authentication model. After the typing patterns are collected, the classifier is built in order to distinguish between the valid user and potential impostors. During authentication, a user provides his ID and password for comparison with the valid user information stored in the system database. If he provides the wrong password, access will not be guaranteed. However, if he provides the correct password, his keystroke dynamics is provided as input to the classifier. If the classifier determines that the user's keystroke dynamics is similar to that of the valid user, access to the system will be granted; else, access will be denied even if he knows the correct password.

Due to their distinctive advantages over systems with other biometric features, KDA systems have been the subject of considerable research [2,7,11,12,17,18,21,24,25,28,33,35–37,43–47,49,50,53–55,60,61,64,66,67]. These studies have confirmed an increase in authentication efficiency in addition to gradual improvement in authentication performance with KDA systems. Further, some companies have successfully developed commercial KDA systems that are now available.[1] However, it should be noted that most current KDA systems have a limited applicability; authentication classifiers are built using short and fixed-length text strings [59,67], e.g., ID and password, typed on the keyboard of a PC during the log-in attempt. Therefore, after a user gains access to the system, it is not possible to determine whether the current user is a valid user who successfully logged in, or an impostor who gained access to a session intentionally or due to improper termination of session by a valid user. In addition, many KDA systems focus on a single input device, i.e., a keyboard of a PC. However, various IT devices that allow network connectivity exist, and each device has a specific input system, which is considerably different from the traditional PC keyboard. Hence, authentication classifiers that perform well with the PC keyboard may not guarantee similar

[1] Some representative commercial systems include Authenware (http://authenware.com), imasicsoftware (http://www.imagicsoftware.com), Biopassword (http://biopassword.com), RSA (http://www.rsa.com), and Delfigo security (http://www.delfigosecurity.com).

performance when used with other input devices. So far, there have been limited work related to KDA systems based on long and free text strings typed on PC keyboards [18,25,50], or other input devices [54,55,61,66].

In this paper, we aim to explore the extendability of KDA for long and free text strings with diverse input devices to fill the vacancy of KDA-related research. Although there has been a large amount of work devoted to KDA systems, there are many practical obstacles when implementing those research results into a real system. For example, since most algorithms were designed for a PC keyboard, their authentication performances were not satisfactory for other input devices. In addition, there has been no systematic analysis for the authentication dependence on the text string length, or for the algorithm selection criteria under certain circumstances. In order to address these practical issues, we first pose three fundamental questions and obtain answers through carefully designed experiments. The three main questions are as follows: (1) Does the authentication performance depend on the type of input device? (2) Does the length of the text affect the authentication performances? (3) Which algorithms are appropriate for user authentication, for the given text length and input device? In order to answer the first question, we collected keystroke data from three input devices by four methods, viz., a traditional PC keyboard, a soft keyboard typed with a stylus pen, and a touch keyboard typed with one hand and two hands. Our hypothesis for the first question was that the authentication performance would depend on the degree of typing freedom, which is a function of the number of fingers (or an equivalent, such as a stylus pen) that could be used for typing. Therefore, we expect that authentication errors are lowest with the traditional PC keyboard, followed by the touch keyboard typed with two hands. We expect that the touch keyboard typed with one hand and the soft keyboard would have the highest authentication error. In order to answer the second question, we conducted authentication performance tests using diverse reference (training or model-building) and test text length combinations. Our hypothesis was that authentication performance would improve when either the reference or text length increased. However, we were unable to determine which one would influence the authentication performance to a greater degree, until we analyzed the experimental results. In order to answer the last question, we employed a total of 12 algorithms, some of which are based on statistical parametric approaches, and the remaining on pattern recognition approaches. Our hypothesis was that with sufficient reference and test keystroke data, more sophisticated algorithms would achieve higher authentication accuracy, while simple statistical methods would be more accurate when limited keystroke data was available. In addition to exploit the responses to these three main questions, we will provide additional meaningful insights by careful observation and analysis of the experimental results.

The rest of the paper is structured as follows. In Section 2, we briefly introduce some representative work that is related to KDA systems with either long and free text strings or various input devices. In Section 3, we explain the process of keystroke data collection, and analyze the summary statistics of the data from each input device. Next, we demonstrate the process for generating reference and test data. In Section 4, we introduce the authentication algorithms briefly with their experimental settings. In Section 5, we analyze the experimental results to determine the responses for each of the main questions. We also provide additional observations or insights, when needed. In Section 6, we state our concluding remarks and discuss future work.

2. Related work

Most previous work related to extending the scope of KDA systems can be divided into two groups, although some work addressed both. Research in the first group involved efforts to build KDA systems using long and free text rather than a short and fixed text, while research in the second group involved efforts to build KDA systems for input devices other than PC keyboards. In this section, we briefly introduce some representative work related to each topic. For more general reviews on KDA systems, ranging from feature extraction to authentication algorithms, please refer to [36,47].

2.1. KDA with long and free text strings

A simple statistical approach for a long and free text-based KDA was explored in [22]. The authors attempted to construct a histogram of down-down time, which is equivalent to the sum of the duration and interval shown in Fig. 1, of the long and free keystroke dynamics of each user. After collecting the keystroke data from each user, a simple approximation is made in order to ensure that the cumulative distribution function for the user has a closed analytic form. For authentication, the overlap area of two histograms was computed. If the overlap area was large, the two keystroke data sets were considered to be obtained from the same user; otherwise, they were considered to be from different users. Although the idea was simple and practical, the length of the text in the experiments was insufficient, and the number of participants was inadequate to generate statistically meaningful results.

R and *A* measures [25] compute the relative and absolute degree of disorder between the typing speed of two users on digraphs, a set of combinations of two consecutive keys. With *R* measure, if two users have a similar order of digraph typing speeds, they will be identified as the same user. *R* measure takes into consideration only the relative order of typing speed but not the absolute typing speed; therefore, in this method, it may be difficult to distinguish two users having a similar order of the digraphs, but a great difference in typing speeds. In order to overcome this limitation, *A* measure compares the absolute typing speed between two users for the digraphs. If two users have similar speeds for the same digraphs, then they are considered as the same user; otherwise, they are considered as different users. Since *R* and *A* measures were one of the most successful authentication methods for long and free text strings, variations of these measures have also been proposed to boost authentication performance [29,42].

A method based on clustering techniques was used to reduce the number of digraphs [50]. The main idea for this work originated from the fact that if the entire digraphs on the PC keyboard were used for authentication, then, very long keystroke data would be required for a statistically meaningful comparison. Therefore, if similar digraphs of a legitimate user can be grouped together, the number of digraphs and the size of keystroke data can be reduced while maintaining authentication performances. Although the experimental results from this research were impressive, it would be difficult to practically apply the methods due to the critical user-specific parameters, e.g., the number of clusters, the threshold for multiclass classification.

Several distance metric-based authentication methods for long and free text were also proposed [32]. This work compared Euclidean distance [7], Manhattan distance [31], Gaussian probability density distance [18,57], *A* measure [25], and the proposed Hamming distance. The experimental results were promising; however, the number of potential impostor data sets greatly outnumbered the number of valid user data sets to achieve an extremely low false acceptance rate. In addition, the KDA system was language-dependent due to the usage of Japanese as the testing language, and pre-defined digraphs, which may depend on the typing language. Further, the length of the text did not seem sufficient because only two hundred keystrokes were collected for the experiment.

In order to overcome the limitations of an individual distance metric, a combination of multiple distance metrics was also proposed ([68]). They combine the Mahalanobis distance and Manhattan distance; the former is used to decorrelate and normalize the keystroke dynamics features, whereas the latter is used to compute the distance in a more standardized new feature space. The experimental results show that the combined distance metric reduces the equal error rate by 16% compare to a single distance metric.

2.2. KDA with various input devices

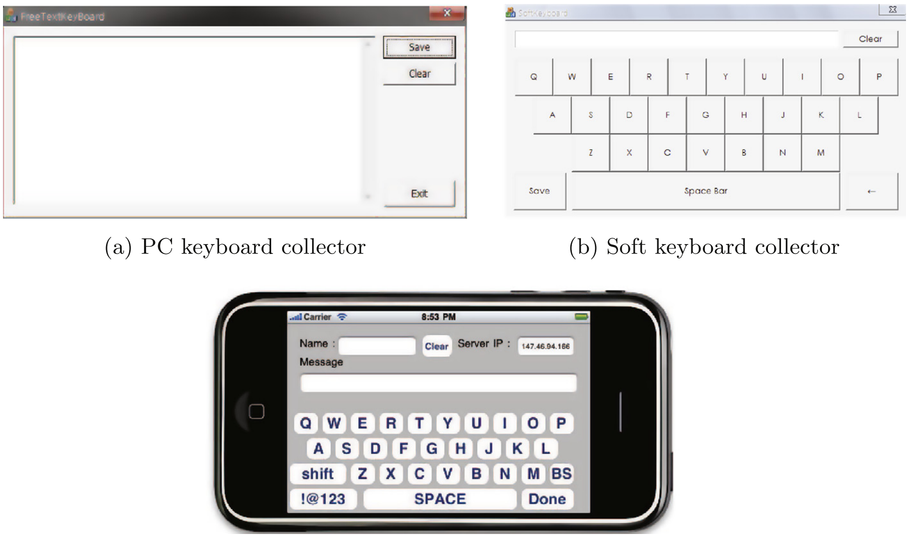
Long and free text-based user authentication with different keyboards was studied in Tappert et al. [54,55], Villani et al. [61], Zack et al. [66]. In the experiments performed in these studies, two keyboard types (desktop and laptop) and two input modes (copy the long text strings and type free text strings) were investigated. The authors also extracted 239 features, including mean and standard deviation of each key sequence, from the keystroke data. Based on the experimental results, it was concluded that the authentication performance was highly reliable when the same type of keyboard was used for both the registration and the test; otherwise, the error rate increased. Although the results were significant, this research had limitations, such as the adoption of binary classification scheme for identification, and the comparison between only two input devices which share the most attributes, i.e., keyboards in desktop and laptop systems.

In [9], user authentication based on fixed alphabetic strings for cellular phones was investigated. When users enrolled their keystroke dynamics for the pre-defined strings, the authenticator generated a representative feature vector. Then, when a new keystroke dynamics was provided, the similarity score between the new keystroke dynamics and the representative vector was computed using various normalization methods. Experimental results showed that an equal error rate (EER) of approximately 13% could be achieved.

KDA systems with a mobile keypad for typing a text message were studied in [14,15,34]. During the experiments, participants provided a pre-defined number of text messages. Then, a feed-forward neural network was trained based on the keystroke data collected, and it achieved an EER of approximately 10%. This research was one of the pioneering studies in which KDA was applied for input devices other than PC-like keyboards; however, it had some limitations. The length of the text message was insufficient and it was extremely difficult to optimize the numerous parameters of the feed-forward neural networks. Further, in recent times, the input system of a mobile phone has been transitioning toward a touch keyboard rather than a keypad.

3. Keystroke dynamics data generation

In order to analyze the device dependence of KDA with long and free text strings, we developed keystroke data collection programs for three different input devices as shown in Fig. 3: a traditional PC keyboard, a soft keyboard, and a touch keyboard. The data structure obtained from these programs is shown in Table 1. Each program captured the down-down time of every keystroke, which is the elapsed time between a key press and the next key press (equivalent to the sum of duration and interval in Fig. 1), when the participants were typing during the program execution. The program for the traditional PC keyboard (Fig. 3(a)) resembles the Notepad program on the Windows operating system. The participants were instructed to type text strings having a minimum length of 3000 characters in this program on their PCs. For the soft keyboard collector (Fig. 3(b)), we simulated a situation in which the users utilized a stylus-pen as input device for a laptop computer or another touch-screen embedded IT device. The participants were requested to visit our laboratory and provide their keystroke data of at least 3000 characters using the laptop computer with the soft keyboard collector program installed. The participants are given enough time to adapt to the stylus-pen before providing their own keystroke data. Some IT devices, such as smartphone or smartpad, provide input systems in which the fingers can be used to type on the keyboard instead of using an electronic device, such as a stylus pen. This environment was simulated by the touch keyboard collector (Fig. 3(c)). We recruited



(a) PC keyboard collector

(b) Soft keyboard collector

(c) Touch keyboard collector

Fig. 3. The keystroke data collector used for different input device types.

Table 1
An example of keystroke dynamics data.

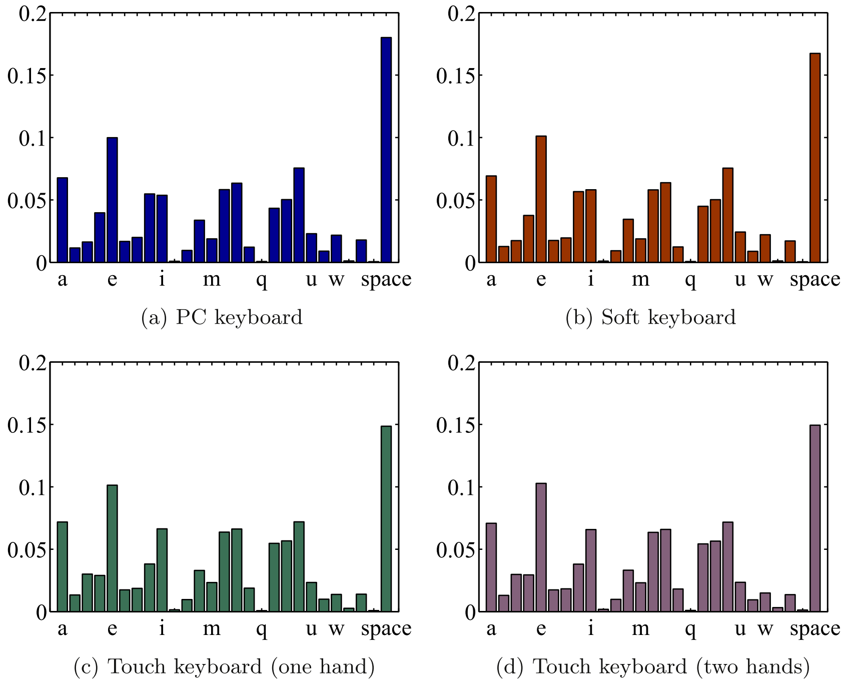
Sequence index	Start key	End key	Down-down time (ms)
1	s	m	250
2	m	c	345
...	...	...	...

participants who had been using an iPhone for more than a year to eliminate possible problems caused by different level of typing expertise. For those participants, we distributed the program with the instruction that two sets of keystroke data, each having a minimum length of 3000 characters, must be provided. The participants were instructed to type the text strings in two ways (using a single hand and using both hands) in order to investigate the difference between one-handed typing and two-handed typing. When using a single hand, it was observed that most participants used either their thumb or index finger, but not both. When using both hands, all participants used the two thumbs for typing text.

A total of 35 participants were involved for the experiments for each input device. Some of them participated in all four experiments. Most participants were graduate students. One-third of them were female while the others were male, with ages ranging from the mid-twenties to the early thirties. Table 2 shows the basic summary statistics of the data collected from the participants for each experiment. As mentioned earlier, it was necessary for every participant to provide greater than 3000 characters of keystroke data excluding special characters (?, !, -, etc.). It was observed that the participants typed an average of approximately 4000 characters with the PC and soft keyboards, and an average of 3600 and 3800 characters with the touch keyboard using one hand and two hands, respectively, during the experiments. Fig. 4 shows the relative frequency of each key from the aggregated keystroke dynamics of all 35 participants. Although slight differences in statistic are observed with the usage of the various input devices, the general trends are similar. First, the spacebar was the most frequently typed key, with frequencies between 14.85% and 18.01%. Apart from the spacebar, four vowels (‘a’, ‘e’, ‘i’, ‘o’) were included in the most frequently typed five keys in all the experiments except the PC keyboard, where only three vowels (‘a’, ‘e’, ‘o’) were included. Among the consonants, ‘t’ and ‘n’ were included in the top five for the PC keyboard, while only ‘t’ was included in the top five for other three experiments. Table 3 shows the five most frequent syllables in the four experiments.

Table 2
The basic summary of four keystroke data sets.

Keyboard type	PC	Soft	Touch (one hand)	Touch (two hands)
No. of participants	35	35	35	35
Avg. text lengths	3953	3994	3596	3809
Max. text lengths	4384	4573	4001	4371
Min. text lengths	3394	3405	3161	3370



(a) PC keyboard

(b) Soft keyboard

(c) Touch keyboard (one hand)

(d) Touch keyboard (two hands)

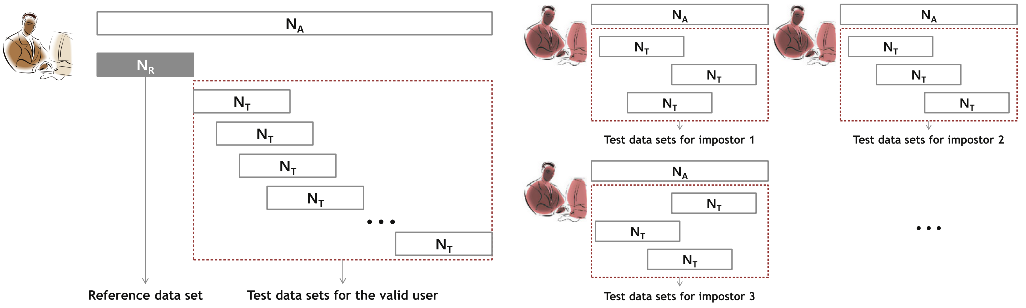
Fig. 4. The relative key frequency of each experiment.

Table 3
The five most frequently typed syllables.

Rank	PC	Soft	Touch (one-hand)	Touch (two-hands)
1	he (4.34%)	he (4.19%)	th (2.67%)	in (2.64%)
2	th (4.09%)	th (3.92%)	in (2.67%)	th (2.62%)
3	an (2.49%)	in (2.48%)	he (2.47%)	he (2.40%)
4	in (2.47%)	an (2.31%)	an (2.31%)	an (2.22%)
5	er (2.13%)	er (2.07%)	er (1.80%)	er (1.84%)
Total	15.52%	14.96%	11.94%	11.73%

Although the rankings differ depending on the experiments, the five most frequent syllables are identical: ‘an’, ‘er’, ‘he’, ‘in’, and ‘th’. Note that the concentration ratio of the top five syllables was higher for the PC keyboard (15.52%) and the soft keyboard (14.96%) than the touch keyboard typed with one hand (11.94%) or two hands (11.73%).

In order to analyze the dependence of authentication performance on the length of the text, we created the reference and test keystroke data sets by varying the text length from 100 to 1000 in step of 100, thus resulting in 100 combinations (10 reference sizes × 10 test sizes). The process for reference and test data generation is illustrated in Fig. 5. Let N_A, N_R and N_T



(a) Reference data construction for a valid user

(b) Test data construction for potential impostors

Fig. 5. The process for reference and test data generation.

denote the number of keystrokes (text length) of the entire, reference, and the test data, respectively. For each user, the first N_R keystrokes were preserved as the reference data set. Note that since the authentication algorithms based on the fixed input variable structure (Group C in the Section 4.3) require a training data set, we constructed a total of 30 keystroke data sets from the reference data set using the bootstrapping technique. Then, for a valid user, from the remaining $N_A - N_R$ keystrokes, 102 test data sets were sampled with the length of N_T. As a result of the overlap between the test sets for the valid user, the length of overlap between any two adjacent test data sets was equalized for fair comparison (Fig. 5(a)). After generation of the reference and the test sets for the valid user, the keystroke data of the remaining 34 participants was used for the creation of test data sets of potential impostors. For each participant, three sets of N_T consecutive keystrokes were randomly selected. By repeating this process, 102 data sets of potential impostors (34 participants × 3 data sets, Fig. 5(b)) were generated. Therefore, with the given reference and test lengths, N_R and N_T, we obtained one reference data set and 102 test data sets of the valid user, and 102 test data sets of potential impostors.

4. Authentication methods

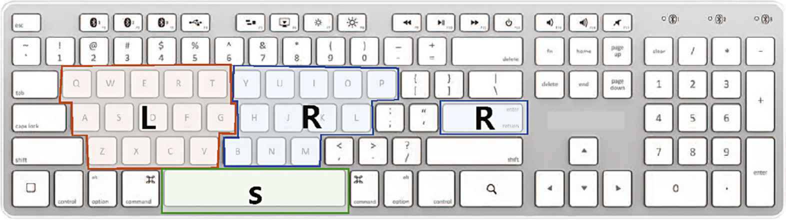
Two types of classification methods are possible although only a legitimate user's typing patterns are provided during registration. One-class classification, known as novelty detection or outlier detection, is available if the classifier is built solely on the basis of data from a valid user [2,28,33,37], while binary or multi class classification can be used if the classifier is built on the basis of data of both a valid user and impostors. Data of impostors can be either synthetically generated or randomly collected from anonymous users [12,24,35,49]. Although binary or multi-class classifiers have higher authentication performances when the potential impostor data sets are well-constructed, they are practically less adaptable than one-class classifiers when the number of system users is very large because impostor data construction for every user would be practically impossible. In addition, the systems have no information on potential impostors, hence, it is more appropriate to use the one-class classification scheme.

We built 12 one-class classifiers for our KDA systems. The classifiers employed in our experiments can be divided into three groups, based on (1) the utilization of key sequence information and (2) the structure of input variables. The comparison among the three groups and membership of each authentication algorithm is summarized in Table 4. As shown in Table 1, our keystroke data consists of the key sequence information (start key and end key), and the elapsed time between two key presses (down-down time). Group A utilizes only the down-down time and ignores the key sequence information. Therefore, for those algorithms in the Group A, the keystroke data is transformed into one-dimensional samples. Then, we can conduct either a parametric or a non-parametric statistical test to compare the sampling distributions between two participants. Group B and Group C utilize the both down-down time and key sequence information, which in turn various machine learning or pattern recognition algorithms can be adopted to distinguish the valid user from the others. The difference between these two groups is the input variable structure; the authentication algorithms in Group B does not fix the input variable structure, but it can be variable according to the reference and test keystroke datasets. The algorithms in Group C, on the other hand, define a fixed input variable structure, and all keystroke data sets are transformed to meet the predefined structure. Fig. 6 shows the input variable structure used in this study. Since the distances between keys are different, the locations of two consecutive can be a major factor that affects the down-down time of the same participant. Thus, we divide the entire keys into three groups (R, L, and S) based on their locations in the qwerty-style keyboard, which is consistently used in our experiment; group L and R include the keys mainly typed by the left and the right hand, respectively, whereas the spacebar, which is usually typed by the thumb of either hand, is taken as an independent group (S). For a set of keystroke data, the eight average down-down times are computed by considering the locations of the start and the end key. Note that although there exist nine possible routes, we ignore the S–S route because it does not happen if one types a text correctly.

Next, we will briefly introduce three statistical tests based on down-down time only, and five pattern recognition techniques based on both the key sequence information and the down-down time. We have developed some of these tests and adopted the others from past research.

Table 4
Attributes of each authenticator group and algorithms belong to each group.

Attribute	Group A	Group B	Group C
Down-down time	Use	Use	Use
Key sequence information	Do not use	Use	Use
Input variable structure	N/A	Flexible	Fixed
Algorithms	M-V Test K–S Test C–M Test	DD *R* measure *A* measure *R* + *A* measure *RA* measure	Gauss Parzen *k*-NN SVDD



Variable	Definition	Description
Var. 1	Avg. L-L	Average down-down time between two keys in the L area
Var. 2	Avg. L-R	Average down-down time from a key (L) to another key (R)
Var. 3	Avg. L-S	Average down-down time from a key (L) to the space bar (S)
Var. 4	Avg. R-L	Average down-down time from a key (R) to another key (S)
Var. 5	Avg. L-L	Average down-down time between two keys in the R area
Var. 6	Avg. L-S	Average down-down time from a key (R) to the space bar (S)
Var. 7	Avg. S-L	Average down-down time from the space bar (S) to a key (L)
Var. 8	Avg. S-R	Average down-down time from the space bar (S) to a key (R)

Fig. 6. The input variable structure for the authentication algorithms in Group C.

4.1. Group A: using only keystroke time

4.1.1. The mean and variance equality test

Here, we assumed that the one-dimensional keystroke dynamics data, which consists of only down-down times, follows the normal distribution. This would be a very strict assumption because the keystroke data of the users may have shapes different from the normal distribution. We adopted this test because it is the quickest and simplest statistical test; therefore, it was used as a baseline authentication performance for other algorithms.

Given a one-dimensional keystroke reference data set K_R and a test data set K_T, let $\bar{\mathbf{x}}_R$ and $\bar{\mathbf{x}}_T$ denote the mean down-down time of the reference participant and the test participant, respectively, while N_R and N_T denote the length of the reference and test data, respectively. Since the normal distribution is determined by the two parameters, the mean (μ) and the standard deviation (σ), we employed Welch's t-test [63] for the mean equality test, and the F-test [30] for the variance equality test. In Welch's t-test, the test statistic t and the degree of freedom $(d.f.)$ are defined as follows:

$$t = \frac{\bar{\mathbf{x}}_R - \bar{\mathbf{x}}_T}{\sqrt{\frac{s_R^2}{N_R} + \frac{s_T^2}{N_T}}}, \quad d.f. = \frac{\left(s_R^2/N_R + s_T^2/N_T\right)^2}{\frac{\left(s_R^2/N_R\right)^2}{(N_R-1)} + \frac{\left(s_T^2/N_T\right)^2}{(N_T-1)}}, \tag{1}$$

where s_R^2 and s_T^2 are the standard deviation of the sample K_R and K_T, respectively. In the F-test, the test statistic F and the degree of freedom $(d.f.)$ are defined as follows:

$$F = \frac{s_R^2}{s_T^2}, \quad d.f. = (N_R - 1, N_T - 1). \tag{2}$$

After conducting these two tests, two significant probabilities are obtained: p_t for the t-test and p_F for the F-test. In order to combine p_t and p_F, we adopt Fisher's method [23] with the following test statistic:

$$\chi^2 = -2(log(p_t) + log(p_F)), \quad d.f. = 4. \tag{3}$$

This statistic follows the chi-squared distribution. Hence, it will be lower when the reference and the test keystroke data are from the same participant than when they are from different participants.

4.1.2. Comparing two empirical distributions

Since the normality assumption in the mean and variance equality tests is extremely rigid in practice, we employed two other statistical methods that does not require the normality condition: Kolmogorov–Smirnov statistic (K–S statistic) [41,52] and Cramér–von Mises criterion (C–M criterion) [1]. In both tests, it was assumed that each participant had a specific down-down time distribution, irrespective of whether it followed the normal distribution. Based on this assumption, we could distinguish one participant from the others by comparing their empirical distributions when sufficient keystroke data was provided. In the K–S test, the empirical distribution functions of the keystroke data sets K_R and K_T are defined as

$$F_{K_R}(x) = \frac{1}{N_R}\sum_{i=1}^{N_R}(I_{X_i} \leqslant x), \tag{4}$$

$$F_{K_T}(x) = \frac{1}{N_T}\sum_{j=1}^{N_T}(I_{X_j} \leqslant x), \tag{5}$$

where $I_{X_i} \leqslant x$ is the indicator function; it is equal to 1 if $X_i \leqslant x$ and equal to 0 otherwise. Then, the K–S statistic is

$$D_{K_R,K_T} = \sup_{\mathbf{x}}|F_{K_R} - F_{K_T}|. \tag{6}$$

Fig. 7(a) illustrates an example of two empirical cumulative distributions and their K–S statistic. The K–S statistic can be interpreted as the maximum difference between two cumulative distributions. Thus, this statistic would be much smaller if two keystroke data sets were collected from the same participant than if they were collected from different participants. Hence, we used the K–S statistic as a discriminant score in our KDA systems.

The C–M criterion is an alternative to K–S statistic for comparing two empirical distributions. In the C–M criterion, the overall difference between two distributions is taken into account rather than a single maximum difference as in the K–S statistic. The C–M criterion is defined as

$$\omega^2 = \int_{-\infty}^{\infty}[F_{K_T}(x) - F_{K_R}(x)]^2 dx. \tag{7}$$

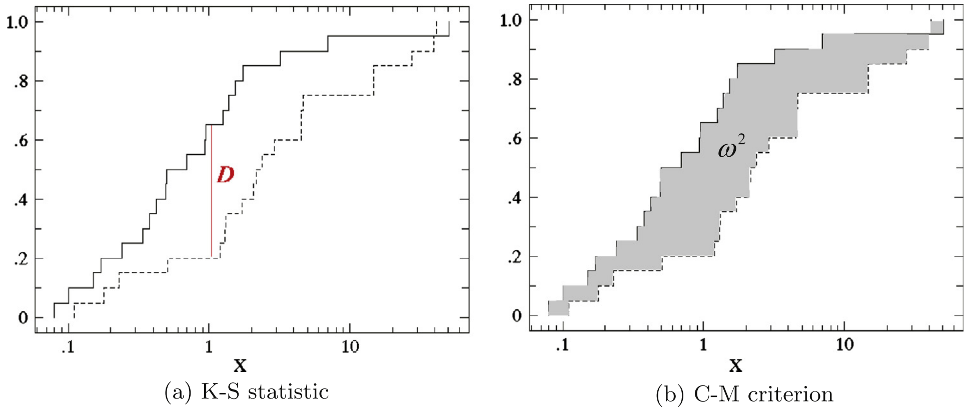
Fig. 7(b) illustrates the C–M criterion of the same empirical distributions in Fig. 7(a). Similar to the K–S statistic, C–M criterion will be larger when two keystroke data sets are collected from different participants than when they are collected from the same participant. The main difference between the K–S statistic and the C–M criterion is that the C–M criterion is able to retrieve information about the overall shape of the differences while the K–S statistic is unable to do so because it only computes the maximum difference. Therefore, we expect the C–M test to achieve higher authentication performances than the K–S statistic.

4.2. Group B: keystroke time, key sequence information, and flexible variable structure

Here, we introduce two families of authentication algorithms based on key sequence information and down-down time with a flexible input variable structure: (1) the distance between two digraph matrices and (2) the R and A measures and their two combinations.

4.2.1. Digraph distance (DD)

When the keystroke data is provided, a 27 × 27 digraph matrix can be constructed, where the row is associated with the start key and the column is associated with the end key. In addition, the element in each cell contains the average down-down time of the corresponding syllable. If the keystroke data is sufficiently long, there would be few empty cells, while the matrix would be sparse if the keystroke data is short. In either case, two participants may share some non-empty cells in their digraph matrices. If two keystroke data sets were collected from the same person, the average down-down times of the common syllables would be very similar; otherwise, they would be different. In order to compare the average down-down times of the common syllables of two participants, we propose the computation of the distance of two digraph matrices **A** and **B** as follows:



(a) K-S statistic

(b) C-M criterion

Fig. 7. An illustrative example of the K–S statistic and C–M criterion.

$$
\begin{aligned}
& d(\mathbf{A},\mathbf{B}) = \sum_{(i,j)\in I}\sqrt{w_{ij}^2(\mathbf{A}_{ij}-\mathbf{B}_{ij})^2}, \\
& I = \{(i,j)|\mathbf{A}_{ij} \neq 0, \mathbf{B}_{ij} \neq 0\},
\end{aligned} \tag{8}
$$

where $\mathbf{A}_{ij}(\mathbf{B}_{ij})$ are associated with the average down-down time of the syllable of the start key i and the end key j of the participant A (B), while I is the set of indices of common non-empty cells of both $\mathbf{A}$ and $\mathbf{B}$. w_{ij} is the weight factor, which is defined as the relative frequency of the common syllable (i,j) such that it emphasizes the syllables typed more frequently compared to the ones typed less frequently. Fig. 8 illustrates an example of digraph distance computation. In this example, values in the parenthesis denote the number of typing and the average down-down time for the corresponding syllable. For example, User R types the syllable 'ac' twice and the average down-down time is 120 ms. Since only two syllables, 'ac' and 'ca', are typed in common, the total typing numbers becomes 9 (=2 + 2+1 + 4) and the weights for 'ac' and 'ca' become 1/3 and 2/3, respectively. Then, the distance between two digraph matrices becomes $\sqrt{\frac{1}{3}\times(120-150)^2+\frac{2}{3}\times(150-100)^2)}$. As the distance increases, the probability that the two keystroke data sets are from different users increases; as the distance decreases, the probability that the two keystroke data sets are from the same user increases.

4.2.2. R and A measures and their combinations

The R and A measures [4,25] determine whether two keystroke data sets are from the same user, based on the 'degree of disorder' (R measure) or the 'ratio of average typing time' (A measure) of common syllables. Fig. 9 illustrates the process of computing the R and A measures using a simple example. Assuming that user $E1$ types the word 'authentication' and user $E2$ types the word 'theoretical', the keystroke data of two users can be collected as shown in Fig. 9(a), where the number preceding each character denotes the hypothetical time in milliseconds when the corresponding key was pressed. The first step in the computation of R and A measures involves the extraction of the common syllables typed by the two users and their corresponding down-down times, i.e., {'ic', 'he', 'th', 'ti', 'ca'} in our example. The second step of R measure is the determination of the degree of disorder for the common syllables in the following manner. First, sort the syllables typed by each user in ascending order of down-down time, and retain the ranking of the syllables. For example, the five syllables are sorted as {'ic', 'he', 'th', 'ti', 'ca'} for user $E1$ and {'th', 'he', 'ca', 'ic', 'ti'} for user $E2$. Next, compute the individual degree of disorder in terms of the difference in the rankings for a syllable between two users. For example, 'ic' is the fastest typed syllable by user $E1$; hence, it has been ranked first. However, it is the fourth-fastest typed syllable by user $E2$, with rank 4. Therefore, the individual degree of disorder of 'ic' becomes 3 (|1–4|). By repeating this process for all common syllables, we can compute the aggregate degree of disorder as shown in Fig. 9(b), in which the aggregate degree of disorder is equal to 8 (2 + 0+2 + 3 + 1). The third step in the computation of the R measure is to normalize the aggregate degree of disorder by the theoretical maximum degree of disorder of the common syllables that can be computed as

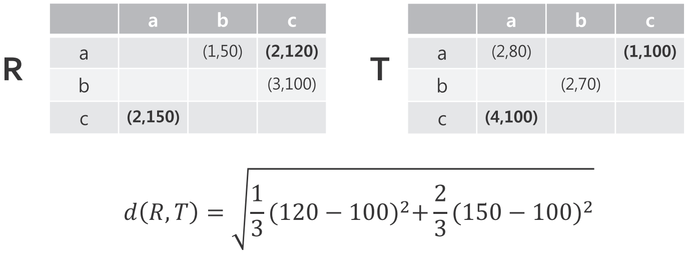
$$
\text{Max. degree of disorder} = \begin{cases} n^2/2 & \textit{if } n \text{ is even}, \\ (n^2-1)/2 & \textit{if } n \text{ is odd}, \end{cases} \tag{9}
$$

where n is the number of common syllables typed by the two users. Finally, the R measure is computed as follows:

$$
R \text{ measure} = 1 - \frac{\text{Aggregated degree of disorder}}{\text{Max. degree of disorder}}. \tag{10}
$$

Since the value of n is 5 in our example, the maximum degree of disorder is equal to 12 $\left(\frac{5^2-1}{2}\right)$; therefore, the R measure is (2 + 0+2 + 3+1)/12 = 2/3. By normalization, we can ensure that the value of the R measure is between 0 and 1. Thus, it can be used irrespective of the number of common syllables.

Although R measure was found to be quite effective in practice, it is unsuitable for some counter examples [25]. Let us assume that two users have extremely different typing speeds. If the rankings of the common syllables are similar, the R measure is high despite the noticeably different typing speeds. In order to compensate for this limitation, A measure first


R

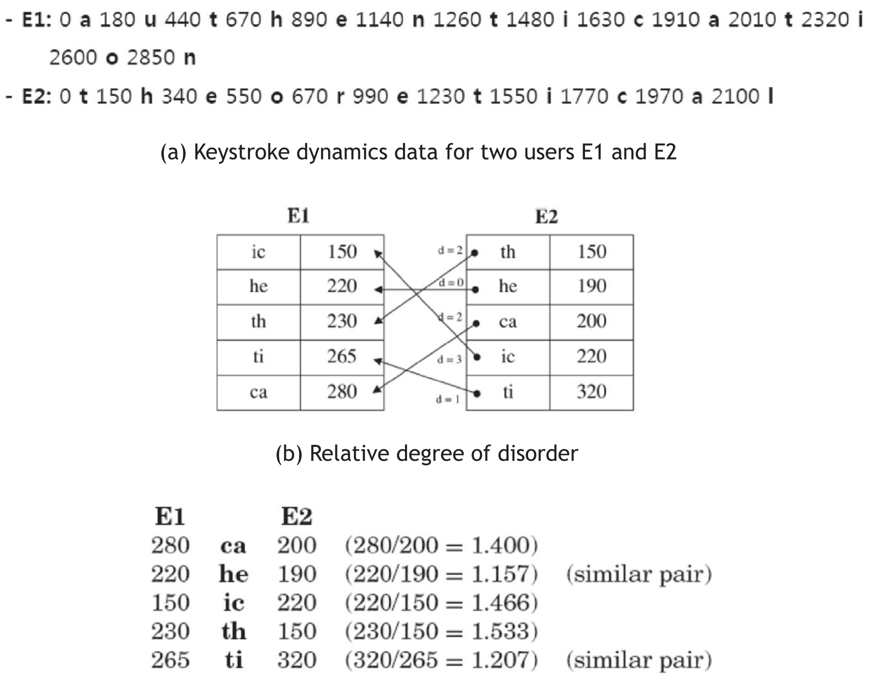
	a	b	c
a		(1,50)	**(2,120)**
b			(3,100)
c	**(2,150)**		

T

	a	b	c
a	(2,80)		**(1,100)**
b		(2,70)	
c	**(4,100)**		

$$
d(R,T) = \sqrt{\frac{1}{3}(120-100)^2+\frac{2}{3}(150-100)^2}
$$

Fig. 8. An example of computing the digraph distance.



- **E1:** 0 **a** 180 **u** 440 **t** 670 **h** 890 **e** 1140 **n** 1260 **t** 1480 **i** 1630 **c** 1910 **a** 2010 **t** 2320 **i** 2600 **o** 2850 **n**
- **E2:** 0 **t** 150 **h** 340 **e** 550 **o** 670 **r** 990 **e** 1230 **t** 1550 **i** 1770 **c** 1970 **a** 2100 **l**

(a) Keystroke dynamics data for two users E1 and E2

(b) Relative degree of disorder

E1		E2		
280	ca	200	(280/200 = 1.400)	
220	he	190	(220/190 = 1.157)	(similar pair)
150	ic	220	(220/150 = 1.466)	
230	th	150	(230/150 = 1.533)	
265	ti	320	(320/265 = 1.207)	(similar pair)

(c) Absolute degree of disorder

Fig. 9. An example of computing the ‘R’ and ‘A’ measures (reprinted from [25]).

computes the ratio of typing speeds between two users for the common syllables, and counts the number of syllables that are typed with a similar speed as follows:

$$A_{count} = \sum_{k=1}^{n} I\left(\frac{max(DD_i^k, DD_j^k)}{min(DD_i^k, DD_j^k)} \leqslant \theta\right), \tag{11}$$

where $DD_i^k\left(DD_j^k\right)$ is the average down-down time of the user $i(j)$ for the kth common syllable, while $I(\cdot)$ is the indicate function that returns 1 when the condition in the parenthesis is satisfied, and returns 0 otherwise. θ is a user-specified cut-off parameter for down-down time similarity. Then, the A measure is computed by dividing A_{count} by n in order to normalize the value. Thus, the A measure is now independent of the number of common syllables:

$$A \text{ measure} = \frac{A_{count}}{n}. \tag{12}$$

In the example shown in Fig. 9(c), if θ is set to 1.4, there are two syllables with ratios between the two down-down times smaller than θ. Thus, the value of A_{count} is 2 and the value of the A measure is 0.4 (=2/5).

The R and A measures can be computed simultaneously with two given keystroke data sets; hence, it is reasonable to combine the R and A measures to strengthen the authentication performance. It was shown that the linear combination of these two measures, as in Eq. (13), can aid in improving the authentication accuracy [25].

$$R + A \text{ measure} = R \text{ measure} + \alpha \times A \text{ measure}, \tag{13}$$

where α is a user-specific combining factor. In this paper, we propose the usage of another combination of the R and A measures based on the product of the two measures, as in Eq. (14).

$$RA \text{ measure} = R \text{ measure} \times A \text{ measure}. \tag{14}$$

The value of both, R and A measures, lie in the range between 0 and 1, and hence the product of both measures is also in the same range. Therefore, in addition to eliminating the need for an additional model parameter for the combination, the measure can also be normalized, which cannot be achieved by a linear combination.

4.3. Group C: keystroke time, key sequence information, and fixed variable structure

Four algorithms are employed for Group C: Gaussian density estimator (Gauss), Parzen window density estimator (Parzen), k-nearest neighbor, and support vector data description (SVDD). They use both key sequence information and down-down time and have a fixed input variable structure.

Gauss ([3]) is the simplest parametric novelty detection method. It assumes a Gaussian distribution for the normal data as shown in Eq. (15).

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]. \tag{15}$$

Thus, training Gauss becomes estimating its two model parameters, $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$, which are the mean vector and the covariance matrix of the normal training data, respectively. Then, whenever a new test instance comes, its normal probability is computed using Eq. (15) with the estimated parameters $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$. If it is high enough, the new instance is classified as the normal class (valid user), otherwise it is classified as outliers (invalid users).

Because Gauss requires a very strong assumption, it is often violated in real datasets, which results in low detection performance. On the other hand, Parzen ([6,20]) does not assume the unimodality in advance but computes the probability based on a mixture of Gaussian kernels centered on the individual training instances as follows,

$$p(\mathbf{x}) = \frac{1}{n}\sum \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left[-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2\sigma^2}\right], \tag{16}$$

where $n, \mathbf{x}_i$, and σ are the number of training instances, the input vector of the ith training instance, and a pre-defined kernel width that is usually obtained by leave-one-out estimation. The main strength of Parzen is that it can generate an arbitrarily complicated decision boundary for the normal class if a sufficiently large dataset is provided.

When a new instance is provided, k-NN finds its k most similar instances based on a certain similarity metric, such as the Euclidean distance. Then, the novelty score is computed by aggregating individual similarity scores of the selected neighbors. Among various similarity combination methods, we adopted the average distance ([26]) to the neighbors as follows,

$$d_{avg}(\mathbf{x}) = \frac{1}{k}\sum_{i=1}^{k}\|\mathbf{x}-\mathbf{x}^i\|, \tag{17}$$

where $\mathbf{x}^i$ is the ith nearest neighbor of the test instance $\mathbf{x}$.

SVDD is a well-known structural risk minimization-based novelty detection algorithm ([56]). SVDD finds a hypersphere with a minimum volume that envelops as many normal instances as possible in the feature space. The optimization problem of SVDD becomes,

$$\begin{aligned} &\min \quad R^2 + C\sum_{i=1}^{n}\xi_i, \qquad (18)\\ &s.t. \quad ||\Phi(\mathbf{x}_i) - \mathbf{a}||^2 \leqslant R^2 + \xi_i, \quad \xi_i \geqslant 0, \forall \mathbf{x}_i, \end{aligned}$$

where $R, \Phi(\mathbf{x}_i)$, and $\mathbf{a}$ denote the radius and the center of the hypersphere, a transformed input data, and the center of the normal class instances in the feature space, respectively. The solution can be found by formulating it as a Wolfe's dual problem and utilizing a kernel trick. When a new instance $\mathbf{x}_n$ is provided, its novelty score can be measured as follows:

$$novelty\ score(\mathbf{x}_n) = R^2 - ||\Phi(\mathbf{x}_n) - \mathbf{a}||^2. \tag{19}$$

In summary, we employed 12 authentication methods in our KDA experiments: the mean and variance equality test (MV test), Kolmogorov–Smirnov statistic (K–S statistic), Cramér–von Mises criterion (C–M criterion), the distance between two digraph matrices (digraph distance; DD), R measures $(R), A$ measures (A), the linear combination of the R and A measures $(R + A)$, the product combination of the R and A measures (RA), Gaussian density estimator (Gauss), Parzen window density estimator (Parzen), k-nearest neighbor detector (k-NN), and support vector data description (SVDD).

4.4. Authentication performance measure

In KDA, two types of misclassification errors exist. False acceptance rate (FAR) is associated with the errors where the system grants access to invalid users, while false rejection rate (FRR) is associated with the errors where the system denies access to valid users. The trade-off between FAR and FRR is that one can be lowered at the cost of the other. For example, an extremely low FRR can be achieved by setting a loose threshold for the algorithms; however, this will result in high FAR. In other words, FRR and FAR depend on the threshold setting as well as the authentication algorithm. As the main goal of our research is to determine the criteria for the selection of appropriate authentication methods under diverse circumstances, we adopt a threshold-independent metric, so called equal error rate (EER) [32,33,36] as the performance measure in the experiments. After training the authentication algorithm, an FAR–FRR relationship figure can be drawn by changing the threshold

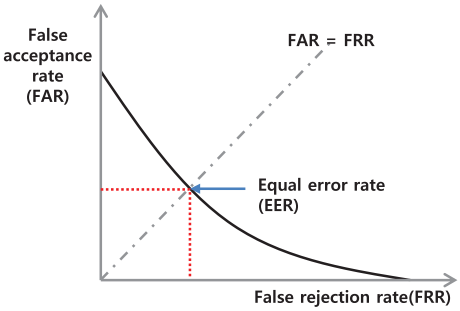


Fig. 10. The equal error rate (EER).

to vary the value of FRR from 0 to 1, as shown in Fig. 10. EER is the error rate when the value of FRR is equal to FAR. The authentication method improves when the value of EER decreases.

5. Experimental results

5.1. KDA performance dependence on the device type

The average EERs of all the authentication methods for each input device and various lengths of reference and test keystroke data sets are summarized in Table 5. Prior to conducting the experiments, we presumed that the degree of typing freedom would be related to the number of fingers (electronic tools) that the participants could use. Therefore, a PC keyboard provided the highest degree of typing freedom because the participants used all their fingers to type the text strings. The next highest degree of freedom was offered by the touch keyboard typed with two hands. It was expected that the soft keyboard and the touch keyboard typed with one hand have the lowest degree of typing freedom because of the use of only a stylus pen (soft keyboard) or a single finger (thumb or index finger) for typing. We also presumed that with an increase in the degree of typing freedom, the authentication performance would improve, thus resulting in a lower EER. The experimental results in Table 5 indicate that while some of our presumptions were correct, others were incorrect. The EERs were the lowest when the participants typed on PC keyboards, which have the highest degree of typing freedom, irrespective of the size of the reference and the test keystroke data. With PC keyboards, the EERs for most reference-test combinations were about 10% lower than the others. Contrary to our presumption, however, on an average, the touch keyboard typed with two hands had slightly higher EERs than the soft keyboard and the touch keyboard typed with one hand on average, specifically when the size of the reference and test data were large, in spite of the fact that the touch keyboard typed with two hands offers a higher degree of typing freedom than the soft keyboard and the touch keyboard typed with one hand. Further, in general, the soft keyboard resulted in lower EERs than the touch keyboard typed with one hand for most cases, although both offer the same degree of typing freedom. It may be possible to provide some explanation for these counter-intuitive results. First, the degree of typing freedom may also be a function of the key size as well as the number of fingers available for typing, and the authentication performance depends on the interaction between these two factors. For example, if the key size is sufficiently large, an increase in the degree of typing freedom would result in lower EERs; however, if the key size is relatively small, an increase in the degree of typing freedom would result in higher EERs. The soft keyboard (implemented on the screen of the laptop computer) has a larger key size than the touch keyboard (implemented on the screen of the smartphone); hence, the EERs of the former were typically lower than the EERs of the latter. In the case of very small key size, an increase in the degree of typing freedom may cause typing errors more often, thus making it difficult to capture the distinctive typing patterns. Thus, the touch keyboard typed with one hand had lower EERs than the touch keyboard typed with two hands because the former has a lower degree of typing freedom. Second, the employed authentication algorithms were unable to record the different attributes between the touch keyboard typed with one hand and with two hands; hence, the EERs were biased toward the touch keyboard typed with one hand. Third, more fundamentally, the participants may have different typing expertise for different input devices. Since all the participants are very familiar with the PC keyboard, it is easier to develop their own typing patterns. On the other hand, they have not experienced the soft and touch keyboards as long as the PC keyboard, it can be hard to develop distinguishable keystroke patterns for these input devices.

5.2. KDA performance dependence on the text length

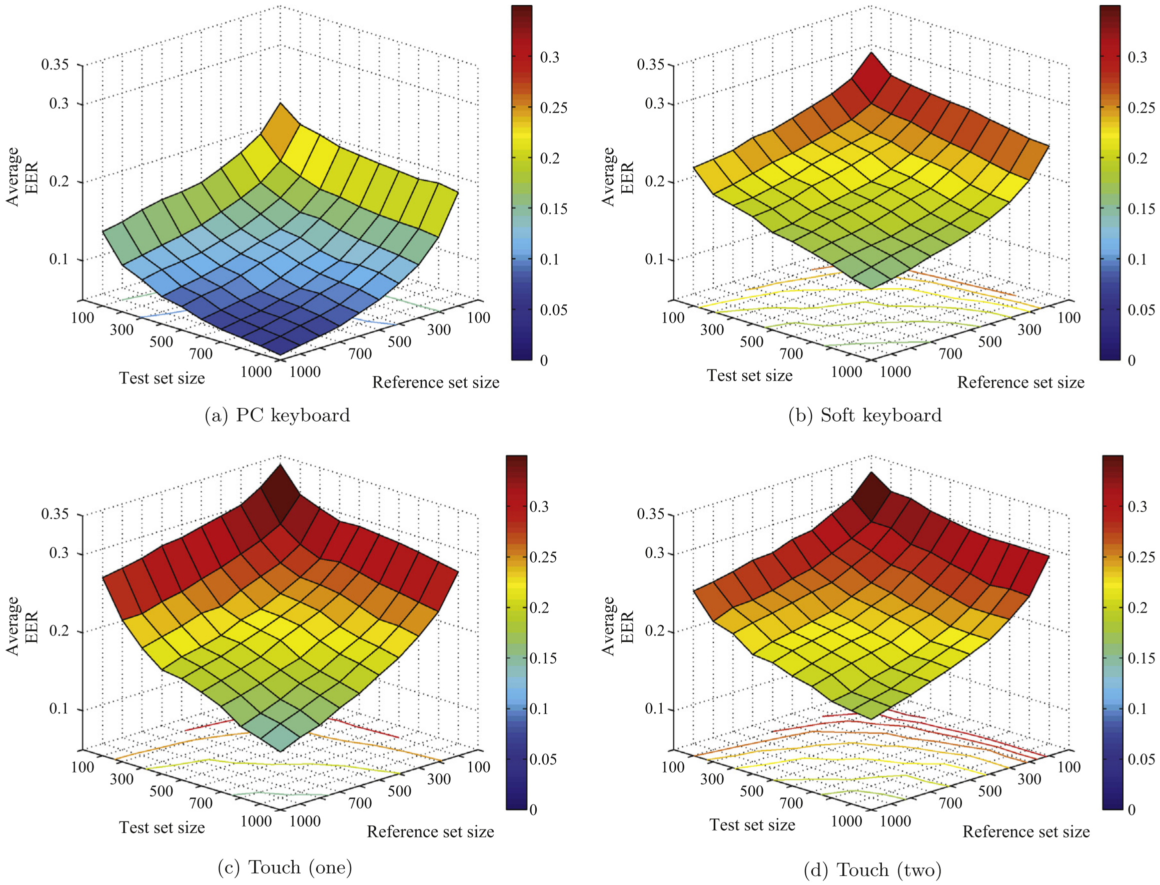
Fig. 11 shows the average EERs of the 35 participants for the 12 authentication methods and the 100 reference-test data set combinations for four experiments. It was observed that, irrespective of the device type, the EERs decreased when either the reference or test length increased. When the reference and test length were the shortest, i.e., 100, the average EERs were the highest for all four experiments: 24.11%, 30.61%, 35.43%, and 34.48% for PC keyboard, soft keyboard, touch keyboard typed with one hand, and touch keyboard typed with two hands, respectively. When either the reference or test size

Table 5
The average EERs of the 35 participants over the 12 algorithms for each device with regard to the size of reference and test data sets.

Reference set size	Device type	Test set size									
		100	200	300	400	500	600	700	800	900	1000
100	PC	**24.10**	**20.88**	**19.25**	**18.04**	**16.97**	**16.54**	**16.30**	**15.76**	**15.00**	**14.53**
	Soft	30.61	28.01	26.78	26.01	25.23	24.50	24.28	23.42	23.04	22.73
	Touch (one)	35.43	33.27	31.76	31.13	30.53	29.89	29.61	28.81	28.30	27.85
	Touch (two)	34.48	31.88	30.62	29.29	28.90	27.96	27.78	27.08	26.54	26.13
200	PC	**22.11**	**18.18**	**16.18**	**14.95**	**13.98**	**13.08**	**12.49**	**12.03**	**11.50**	**10.97**
	Soft	28.40	25.56	24.28	23.55	22.90	22.28	21.85	21.23	20.59	20.08
	Touch (one)	32.35	29.33	27.99	27.04	26.09	25.36	24.49	23.99	23.24	23.18
	Touch (two)	32.73	29.64	28.26	27.11	25.95	25.40	24.68	24.22	24.06	23.02
300	PC	**21.55**	**17.00**	**14.99**	**13.70**	**12.58**	**11.85**	**11.10**	**10.71**	**10.42**	**9.89**
	Soft	27.92	24.99	23.37	22.78	22.19	21.59	20.74	20.21	19.67	19.42
	Touch (one)	31.33	27.51	25.89	24.92	23.91	22.62	22.51	21.65	21.03	20.37
	Touch (two)	32.61	29.60	27.45	26.16	24.92	24.44	23.87	23.31	22.83	22.20
400	PC	**20.76**	**16.12**	**13.91**	**12.55**	**11.77**	**10.92**	**10.09**	**9.48**	**9.03**	**8.50**
	Soft	27.59	24.77	23.01	22.27	21.51	20.86	20.38	19.52	19.00	18.74
	Touch (one)	30.45	26.43	24.82	23.48	22.30	21.65	20.44	19.75	19.26	18.34
	Touch (two)	31.79	28.27	26.41	24.89	23.89	22.78	22.23	21.68	21.17	20.32
500	PC	**20.45**	**15.89**	**13.87**	**12.21**	**11.29**	**10.50**	**9.67**	**8.95**	**8.58**	**7.98**
	Soft	27.30	24.07	22.21	21.01	20.18	19.63	19.12	18.63	17.86	17.50
	Touch (one)	30.58	26.78	24.64	23.21	22.12	21.09	20.20	19.31	18.68	17.97
	Touch (two)	31.31	28.05	26.20	24.34	23.42	22.32	21.83	21.28	20.79	20.04
600	PC	**20.17**	**15.40**	**13.18**	**11.34**	**10.44**	**9.40**	**8.45**	**8.01**	**7.54**	**7.13**
	Soft	27.09	23.59	21.93	20.71	20.04	19.26	18.71	18.22	17.59	17.10
	Touch(one)	30.27	26.22	24.24	22.79	21.68	20.59	19.26	18.44	17.74	17.03
	Touch(two)	30.85	27.35	25.29	23.57	22.88	21.66	21.27	20.63	19.93	19.11
700	PC	**19.97**	**15.11**	**12.75**	**10.82**	**9.88**	**8.81**	**8.13**	**7.45**	**7.06**	**6.63**
	Soft	26.60	23.22	21.61	20.07	19.44	18.65	17.87	17.35	17.00	16.44
	Touch (one)	30.00	25.56	23.42	21.60	20.50	19.09	18.33	17.37	16.61	15.96
	Touch (two)	30.60	26.83	24.65	23.17	22.33	21.12	20.54	19.97	19.27	18.53
800	PC	**19.72**	**14.97**	**12.54**	**10.99**	**9.62**	**8.90**	**7.94**	**7.35**	**6.76**	**6.35**
	Soft	26.16	22.65	20.57	19.37	18.42	17.71	17.02	16.63	16.23	15.70
	Touch (one)	29.59	25.17	22.50	20.90	19.61	18.17	17.37	16.10	15.09	14.16
	Touch (two)	30.65	26.12	24.03	22.38	21.23	20.20	19.40	18.78	17.80	17.37
900	PC	**19.99**	**14.90**	**12.26**	**10.80**	**9.44**	**8.26**	**7.59**	**7.10**	**6.50**	**6.04**
	Soft	25.67	22.15	20.09	18.80	17.86	17.17	16.58	16.07	15.40	14.74
	Touch (one)	29.08	24.71	22.42	20.47	19.14	17.55	16.61	15.49	14.53	13.35
	Touch (two)	30.64	26.39	24.01	22.22	21.24	20.27	19.24	18.25	17.62	16.97
1000	PC	**19.57**	**14.57**	**11.81**	**10.20**	**8.87**	**7.95**	**7.03**	**6.54**	**6.11**	**5.64**
	Soft	25.51	21.90	19.84	18.51	17.40	16.62	16.01	15.15	14.59	14.10
	Touch (one)	28.61	24.27	21.88	19.89	18.43	16.86	15.75	14.35	13.54	12.42
	Touch (two)	30.62	26.24	24.05	22.21	21.10	20.06	19.15	18.20	17.41	16.62

increased, the average EERs decreased; thus, the lowest EERs were reported with the longest reference and test length, i.e., 1000: 5.64%, 14.10%, 12.42%, and 16.62% for PC keyboard, soft keyboard, touch keyboard typed with one hand, and touch keyboard typed with two hands, respectively.

Based on these experiments, the next logical step is to determine whether reference size or test size is more effective in lowering the EERs. Fig. 12 depicts the average EERs for four experiments with fixed length of either the reference ((a), (c), (e)) or the test ((b), (d), (f)) keystroke data. For PC keyboard, the average EERs decreased by 9.58%p, 12.48%p, and 13.93%p for the fixed reference lengths 100, 500, and 1000, respectively, as the test length increased from 100 to 1000. However, the average EERs decreased by 4.54%p, 8.11%p, and 8.89%p for the fixed test lengths 100, 500, and 1000, respectively, as the reference length increased from 100 to 1000. Based on these results, it was determined that, for a PC keyboard, the test length affected authentication performance more than the reference length. This observation would be very useful in practice because users may psychologically resist providing a large amount of keystroke dynamics data when they enroll in the system. Based on our observation, this burden can be relieved that the users could provide a minimum number of keystrokes during the enrollment for creation of the authentication model. The model can then be enhanced by collecting additional keystroke data from the users when they are connected to the system.



(a) PC keyboard

(b) Soft keyboard

(c) Touch (one)

(d) Touch (two)

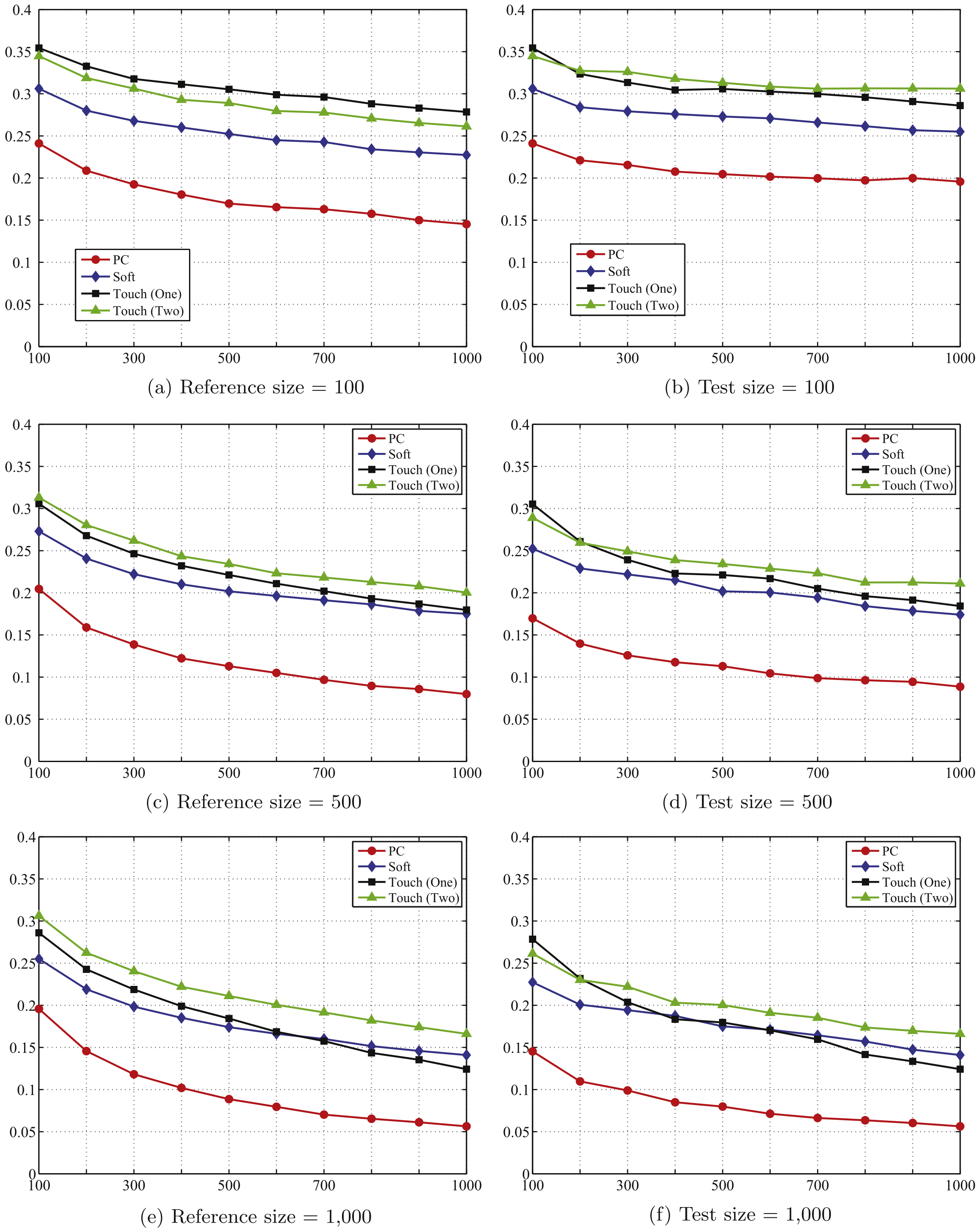
Fig. 11. The average EER trends of the 35 participants for the 12 authentication methods and 100 reference-test data set combinations.

For the soft keyboard, the average EERs decreased by 7.88%p, 9.80%p, and 11.41%p for the fixed reference lengths 100, 500, and 1000, respectively, as the test length increased from 100 to 1000. However, the average EERs decreased by 5.10%p, 7.82%p, and 8.63%p for the fixed test lengths 100, 500, and 1000, respectively, as the reference length increased from 100 to 1000. The observations for the soft keyboard were similar to those for the PC keyboard. Increasing the test length affected the authentication performance to a greater extent than increasing the reference length. However, the degree of EER improvement after increasing the test length was a little lower for soft keyboard than for the PC keyboard, while the degree of EER improvement after increasing the reference length was a little higher for soft keyboard than for the PC keyboard. Therefore, the difference in the EER percentage decrease for the same values of fixed test length and fixed reference length are lower than the corresponding values obtained when the PC keyboard was used.

For touch keyboard typed with one hand, the average EERs decreased by 7.59%p, 12.61%p, and 16.19%p for the fixed reference lengths 100, 500, and 1000, respectively, as the test length increased from 100 to 1000. However, the average EERs decreased by 6.83%p, 12.10%p, and 15.43%p for the fixed test lengths 100, 500, and 1000, respectively, as the reference length increased from 100 to 1000. Thus, unlike the PC or soft keyboard, the differences in the EER percentage decrease between the test length and reference length pairs in the same order above are very small and they are not statistically significant ($\alpha = 0.05$). This implies that it cannot be guaranteed whether the reference length or the test length was more influential on the authentication performance than the other.

For touch keyboard typed with two hands, the average EERs decreased by 8.35%p, 11.27%p, and 14.00%p for the fixed reference lengths 100, 500, and 1000, respectively, as the test length increased from 100 to 1000. However, the average EERs decreased by 3.86%p, 7.80%p, and 9.51%p for the fixed test lengths 100, 500, and 1000, respectively, as the reference length increased from 100 to 1000. The EER decrease trends were very similar to the trends observed with the PC keyboard and the soft keyboard; the test length was more effective than the reference length in decreasing the EER.

In summary, the average EERs decreased with an increase in either the reference length or the test length. The test length seemed to have a greater effect than the reference length on the authentication performance for all experiments except the touch keyboard typed with one hand; the reference and test lengths had a statistically indifferent influence on the authentication performance.



(a) Reference size = 100

(b) Test size = 100

(c) Reference size = 500

(d) Test size = 500

(e) Reference size = 1,000

(f) Test size = 1,000

Fig. 12. The trends of the average EERs of the 12 authentication methods over 35 participants for each experiment when either the size of the reference or the test data is fixed.

5.3. KDA performance variations among the algorithms

In order to compare the overall authentication performance among the authentication methods, the average EERs were aggregated over all reference and test length combinations, as shown in Table 6. The EERs with an asterisk (∗) were statistically lower than those without it by a significant level of 0.05, while those with asterisks were not statistically different. In

Table 6
The mean and the standard deviation of the EERs (%) over all reference-test length combinations for each authentication methods and experiment.

Group	Authentication method	PC	Soft	Touch (one)	Touch (two)
Group A	MV	18.09 (3.60)	21.27 (3.46)	25.12 (4.38)	28.59 (2.52)
	K–S	9.67 (3.03)	25.17 (2.78)	24.55 (4.04)	28.42 (2.29)
	C–M	10.42 (3.17)	**15.29*** (5.58)	18.78* (5.07)	**18.85** (5.85)
Group B	DD	16.43 (5.80)	20.18 (4.00)	23.15 (5.95)	23.01 (5.11)
	R	12.83 (5.59)	28.12 (4.39)	31.48 (5.81)	29.47 (4.93)
	A	10.10 (4.67)	21.47 (3.62)	25.24 (4.64)	27.35 (4.58)
	$R + A$	**7.87*** (4.26)	19.67 (3.97)	23.36 (5.59)	24.39 (5.02)
	RA	8.00* (4.31)	19.85 (4.00)	23.31 (5.51)	24.49 (4.94)
Group C	Gauss	16.06 (5.91)	20.14 (3.79)	20.90 (6.06)	22.86 (4.11)
	Parzen	12.31 (4.73)	19.64 (3.10)	18.67* (5.68)	20.33 (4.06)
	k-NN	12.36 (4.71)	19.65 (3.14)	**18.64*** (5.60)	20.28 (4.15)
	SVDD	12.63 (4.29)	20.41 (2.93)	18.82* (5.38)	20.68 (3.98)

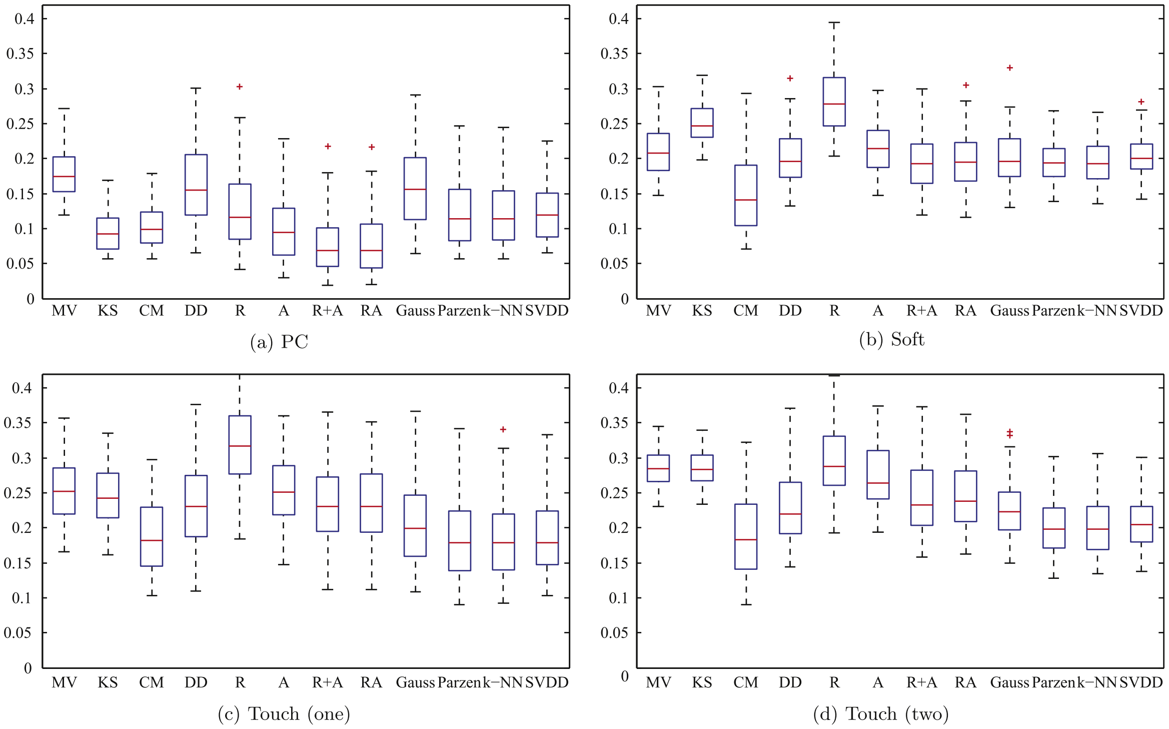
most cases, the $R + A$ measure resulted in the lowest EER for the PC keyboard, while the C–M criterion resulted in the lowest EER for both the soft keyboard and the touch keyboard typed with two hands. Three methods belonging to Group C, i.e., Parzen, *k*-NN, and SVDD resulted in lower error than the other methods for the touch keyboard typed with one hand. For the PC keyboard, the $R + A$ measure was the best, followed by the *RA* measure with no statistical difference. The K–S statistic and C–M criterion were placed next, while the MV test, DD, and Gauss reported high EERs, which were greater than twice the best EER. For the soft keyboard, the C–M criterion resulted in the best EER followed by the DD and four methods in the Group C. Unlike the PC keyboard, *R* measure resulted in higher EER than the *A* measure or the two combinations of the *R* and *A* measures. For the touch keyboard typed with one hand, it was found that the algorithms in Group C, except the Gauss, resulted in similar performances and their EERs are statistically lower than the other algorithms with only one exception (C–M). Similar to the soft keyboard, the C–M criterion reported the lowest EER for the touch keyboard typed with two hands. Although Parzen, *k*-NN, and the SVDD were not found to be the best, they also reported comparable authentication performances with the C–M criterion.

Note that unlike the C–M criterion, K–S statistic resulted in relatively high EERs for the soft and touch keyboards, although it reported good performance for the PC keyboard. This is because the K–S statistic performs well when the shape of the distribution is similar to the normal distribution, which is not a necessary condition for the C–M criterion. In our experiments, the distribution of the down-down times collected from the PC keyboard was similar to the normal distribution; thus, both the K–S statistic and C–M criterion resulted in similar EERs. On the other hand, the down-down times collected from the other input devices did not follow the normal distribution, but exhibited some skewness or irregular patterns; hence, a good discrimination criterion was produced by the C–M criterion but not by the K–S statistic.

Fig. 13 depicts the box plots of the EERs over all reference and test length combinations for every authentication method for each experiment. For the PC keyboard, the MV test, K–S statistic, and C–M criterion showed low variations compared to the others. As mentioned earlier, the $R + A$ measure and *RA* measure resulted in a similar EER for every combination; their box plots also looked very similar. For the soft keyboard, it is noticeable that the EER variations of Parzen, *k*-NN, and SVDD are smaller than those for the other types of input device. We should note that the C–M criterion had the largest variations for the soft and touch keyboards. The reason for this result is that although the average EERs were the lowest with the C–M criterion when the reference and the test length were small, the average EERs even decreased most rapidly as either the reference length or the test length increased.

The average EERs of the 35 participants with some representative reference and test length combinations for the PC keyboard, soft keyboard, touch keyboard typed with one hand, and touch keyboard typed with two hands are shown in Table 7–10, respectively. The numbers with bold face are the lowest average EER among the 12 authentication methods for the corresponding reference and test lengths, while the numbers with an asterisk ($*$) denote that the corresponding method has lower average EER than the other methods at the significant level of 0.05. For the PC keyboard, as shown in Table 7, the K–S statistic was the best method when either the reference or the test length was relatively small. When the reference length is 100, the K–S statistic resulted in the lowest average EER compared to other methods, irrespective of the test length. Further, when the reference length increased to 500, the K–S statistic remained the best for small test lengths. When the reference and test lengths increased, however, the $R + A$ measure and *RA* measure outperformed the other methods. Specifically, when a sufficient amount of keystroke data was available, e.g., 1000 keystrokes for each user, the $R + A$ measure and *RA* measure were able to distinguish potential impostors from the valid user with very low errors (less than 2%).

The authentication ability of the 12 methods for the soft keyboard and touch keyboard typed with two hands appeared similar to each other with a few exceptions. First, one of the algorithms in the Group C was found to be the best method when the reference length is short; *k*-NN and Parzen reported the lowest EER for the soft keyboard when the reference length is 100, whereas SVDD and *k*-NN reported the lowest EER for the touch keyboard typed with two hands for the same reference length. As the reference length increased, however, the C–M criterion resulted in the lowest EER in most cases for the both types of input device. For the touch keyboard typed with one hand, the best authentication algorithms with regard to the



(a) PC

(b) Soft

(c) Touch (one)

(d) Touch (two)

Fig. 13. The box plots of the EERs over all reference-test length combinations for each authentication method and experiment.

Table 7
The average EERs of the 35 participants along with each authentication method for some representative reference-test length combinations when typing with a PC keyboard.

No. Reference	No. Test	Group A			Group B					Group C			
		MV	K–S	C–M	DD	R	A	R + A	RA	Gauss	Parzen	*k*-NN	SVDD
100	100	27.17	**16.95***	17.90*	30.11	30.34	22.80	21.74	21.60	29.08	24.71	24.43	22.49
	500	20.56	**11.82**	11.90	22.21	19.55	16.75	13.70	13.73	23.84	16.95	17.34	15.32
	1000	17.65	**9.30**	9.78	17.90	16.13	15.80	11.18	11.74	23.17	13.56	15.29	12.89
500	100	25.18	**15.77**	16.39	26.86	22.35	17.82	15.83	15.80	23.98	21.37	21.76	22.41
	500	17.17	9.05	9.83	15.38	11.65	8.52	**7.28**	7.37	14.40	11.57	11.62	11.74
	1000	14.01	6.58	6.47	10.70	8.01	6.33	4.51	**4.31**	10.87	7.84	7.76	8.38
1000	100	25.32	15.18	16.86	26.69	20.87	16.83	**14.06**	**14.06**	22.16	20.59	20.70	21.51
	500	16.02	7.82	9.08	11.51	8.18	5.80*	**4.06***	4.20*	11.32	9.10	9.30	10.00
	1000	11.90	5.83	5.71	6.58	4.12	3.00*	**1.90***	2.04*	7.96	5.63	5.97	7.00

Table 8
The average EERs of the 35 participants along with each authentication method for some representative reference-test length combinations when typing with the soft keyboard.

No. Reference	No. Test	Group A			Group B					Group C			
		MV	K–S	C–M	DD	*R*	*A*	*R + A*	*RA*	Gauss	Parzen	*k*-NN	SVDD
100	100	30.31	31.88	29.33	31.51	39.44	29.78	30.00	30.53	33.03	26.81	**26.58**	28.15
	500	26.58	29.13	23.39	23.11	33.78	25.43	24.57	24.82	26.25	**21.76**	21.85	22.07
	1000	25.01	26.92	21.37	**18.80**	31.54	22.77	21.74	21.99	23.78	19.47	20.11	19.27
500	100	25.74	28.91	**24.34**	28.35	35.35	28.04	26.53	27.34	26.13	25.63	25.13	26.13
	500	19.78	24.62	**13.11***	20.28	26.64	20.67	18.63	18.49	19.55	19.80	19.52	21.12
	1000	18.38	22.86	**10.22***	15.43	24.12	17.56	15.18	15.41	18.60	17.20	16.92	18.18
1000	100	23.70	26.75	**22.04**	26.36	33.19	27.34	24.73	24.96	24.15	24.31	23.89	24.71
	500	17.00	22.24	**10.28***	18.07	23.56	18.32	16.11	16.27	16.05	16.61	16.72	17.62
	1000	14.71	19.86	**7.06***	13.28	20.36	14.76	11.96	11.60	13.75	14.12	13.56	14.17

Table 9
The average EERs of the 35 participants along with each authentication method for some representative reference-test length combinations when typing with the touch keyboard with one hand.

No. Reference	No. Test	Group A			Group B					Group C			
		MV	K–S	C–M	DD	*R*	*A*	*R* + *A*	*RA*	Gauss	Parzen	*k*-NN	SVDD
100	100	35.69	33.56	**29.78**	37.62	42.66	35.99	36.58	35.15	36.61	34.15	34.06	33.36
	500	32.52	30.64	**24.54**	30.92	41.23	31.34	31.09	31.76	31.04	27.76	27.06	26.44
	1000	30.31	28.68	**22.02**	26.36	38.94	27.09	27.84	27.93	32.21	25.10	23.67	24.01
500	100	30.70	28.82	27.98	32.44	38.96	32.77	31.09	31.12	28.85	**27.68**	28.29	28.24
	500	25.07	23.75	17.65	23.59	32.10	24.37	23.22	23.11	19.24	**17.54**	17.93	17.84
	1000	22.02	21.06	13.45	17.39	26.78	20.87	18.96	18.54	16.50	**13.25**	13.50	13.31
1000	100	28.38	**25.29**	25.77	30.73	36.19	32.69	29.47	29.69	25.99	25.32	26.41	27.34
	500	22.13	20.31	15.29	19.50	25.43	22.38	18.40	18.60	15.07	**13.67**	15.41	15.01
	1000	16.61	16.16	**10.36**	10.98	18.43	14.76	11.18	11.20	10.90	8.96	9.19	10.28

Table 10
The average EERs of the 35 participants along with each authentication method for some representative reference-test length combinations when typing with the touch keyboard with two hands.

No. Reference	No. Test	Group A			Group B					Group C			
		MV	K–S	C–M	DD	*R*	*A*	*R* + *A*	*RA*	Gauss	Parzen	*k*-NN	SVDD
100	100	34.45	33.95	32.18	37.09	41.68	37.42	37.31	36.25	33.78	29.97	30.03	**29.61**
	500	30.08	30.70	25.41	28.91	37.39	32.16	31.82	32.02	27.87	23.89	23.47	**23.08**
	1000	28.12	29.55	23.45	26.11	33.89	29.55	28.77	28.46	24.57	20.78	**19.47**	20.81
500	100	32.35	31.32	29.30	31.51	37.17	35.63	32.46	32.52	29.78	**27.56**	27.98	28.15
	500	28.60	28.49	**17.68***	21.93	28.91	27.20	23.25	23.22	20.98	20.03	20.03	20.78
	1000	26.22	26.81	**13.25***	17.84	25.83	23.31	19.64	20.00	18.63	16.83	16.02	16.11
1000	100	31.85	30.50	29.47	29.94	33.75	36.08	31.76	32.32	28.74	**27.42**	27.98	27.59
	500	26.50	26.72	**13.78***	19.41	26.05	25.52	21.12	20.92	19.78	17.90	17.42	18.12
	1000	23.05	24.17	**8.99***	14.45	19.30	19.58	15.77	16.41	16.33	12.83	14.68	13.81

reference length are reversed; the C–M criterion reported the lowest EERs when the reference length is short (100), but Parzen was the best with longer reference lengths (500 & 1000) for most cases.

From a practical point of view, the KDA system for non-traditional input device, i.e., soft keyboard and touch keyboard, is not very accurate when the reference and test lengths are very short, e.g., 100, because the average EER is around 30%. However, when sufficient keystroke data is provided, e.g., more than 500, the error rates are lower than 15%. This percentage would be tolerable because users would accept one or two denials of access per ten trials knowing that it would successfully prevent most illegal access trials.

Focusing on the algorithm groups, it cannot be concluded that the algorithms in one group are always better than the algorithms in other groups. The authentication performance depends on both input device type and the length of the reference and the test keystrokes. However, it is rather clear that which algorithm is better than the others within each group. Among the algorithms in the Group A, K–S statistic performed well for the PC keyboard with a short reference length, whereas C–M criterion was the best for most of the other cases. Among the algorithms in the Group B, two combinations of the *R* and *A* measures generally performed better than a single *R* or *A* measure. DD was not found to be as effective as *R* and *A* measure-based authenticators, especially for the PC keyboard. Among the algorithms in the Group C, Parzen generally resulted in a lower EER than the others but the differences are not statistically significant except the Gauss. Since the Gauss made a strong assumption on the distribution of the normal class, it is often violated in practice, which resulted in relatively poor authentication performances.

It must be mentioned that, in contrast to the PC keyboard, a method without key sequence information (C–M criterion) outperformed most methods with key sequence information, although there was a sufficient number of reference and test keystrokes. One possible cause for this behavior is that key sequence information would be useful only when the degree of typing freedom is relatively high. However, when users type with a stylus pen on the soft keyboard or with their fingers on the touch keyboard, the degree of typing freedom is much lower than typing on the PC keyboard with all fingers. Another interesting observation is that *R* and *A* related measures were not appropriate for input systems other than the PC keyboard because they were originally designed for keystroke data collected through the PC keyboard. One possible solution to this problem is to design an effective input variable structure because the algorithms in Group C generally performed better than the *R* and *A* related measures for the soft and touch keyboards despite of the simple input variable structure.

In summary, based on our experiments, we determine that the authentication performance depended on the type of input device. Current authentication methods generally work well with the traditional PC keyboard, while their error rates were higher with the other input devices. Secondly, authentication performance was also dependent on the length of both, the reference and test data sets. If either of them increased, the authentication error decreased. It was also found that, in general, the length of the test data had a greater effect on the authentication accuracy than the length of the reference data. Finally, when either the reference or the test length was small, the K–S statistic was the best among the authentication algorithms, while the $R+A$ and RA measures were the best with sufficient keystroke data in the case of PC keyboard. For the soft and touch keyboard, the C–M criterion, which measures the overall difference between two empirical distributions, and novelty detectors based on the fixed input variable structure, such as Parzen, *k*-NN, and SVDD, performed better than the other algorithms for most reference and test length combinations.

6. Conclusion

In this paper, as a ground work for the extension of KDA coverage, we systematically analyzed the critical issues on KDA system based on long and free text strings with various types of input device. To do so, we first formulated three main questions: (1) Does the authentication performance depend on the type of input device? (2) Does the length of the text affect the authentication performances? (3) Which authentication algorithms are appropriate for certain conditions (input device and text lengths)? In order to obtain practical answers to the questions, we carefully designed the experiments and made the following observations based on the experimental results. First, current authentication methods worked very well with the PC keyboard as input device; however, they require customization when used with other input devices. Second, authentication performance was highly dependent on the size of both reference and test keystrokes. If only one factor should be increased, then, increasing the length of test keystrokes will reduce the authentication error to a greater degree than increasing the reference keystrokes. Third, with the $R+A$ or RA measures, an error rate of nearly zero could be achieved for the PC keyboard when a sufficient amount of keystroke data was provided; however, the error rate increased with other input devices. For the other three input devices, the C–M criterion or one of the three novelty detectors based on the fixed input variable structure, i.e., Parzen, *k*-NN, and SVDD, was found to be the best model for most of reference-test combinations.

So far, our research can be regarded as a basis for the development of KDA systems using long and free text strings from various input devices. Hence, various topics for future research in this area are possible. Firstly, although we achieve very impressive authentication accuracy for the keyboard with sufficient keystroke data, the authentication performances using a few keystroke data with the PC keyboard and overall authentication performance with other input devices should be improved. There are three possible methods to pursue this. First, we only used the down-down time to maintain the data consistency among the different input devices. However, it can be possible to capture more segmented typing activities, e.g., duration (the elapsed time between pressing a key and releasing the key) and interval (the elapsed time between releasing a key and pressing another key). Therefore, the authentication performance can be improved by collecting more segmented keystroke dynamics. Second, more effective derivative features in conjunction with segmented typing activities should be created based on the key sequence information and its down-down time to enhance the performance of the existing authentication methods. Third, other statistical or machine learning algorithms should be tested. Secondly, nowadays, many smart devices provide the assistant mechanisms such as T9 to improve the typing efficiency, which were not considered in the current experiments. Therefore, we should investigate how those helping mechanisms affect the KDA performance and develop customized authentication algorithms for such mechanisms. Thirdly, an authentication method independent of the input device or an integrated KDA framework incorporating all input devices should be designed. This would eliminate the need for users to change input devices that they could use to connect to the network system. Finally, we can also think of an authentication system to incorporate other behavioral biometrics, such as mouse gestures or finger gestures on the touch screen, into long and free text string-based KDA system.

Acknowledgments

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT, & Future Planning (NRF-2014R1A1A1004648).

References

[1] T.W. Anderson, On the distribution of the two-sample Cramér–von Mises criterion, Ann. Math. Stat. 33 (1962) 1148–1159.
[2] L.C. Araujo, L.H.S. Sucupira Jr., M.G. Lizarraga, L.L. Ling, J. Yabu-Uti, User authentication through typing biometrics features, IEEE Trans. Security Privacy 53 (2005) 851–855.
[3] V. Barnett, T. Lewis, Outliers in Statistical Data, Wiley and Sons, New York, NY, USA, 1994.
[4] F. Bergadano, D. Gunetti, C. Picardi, User authentication through keystroke dynamics, ACM Trans. Inform. Syst. Security 5 (2002) 367–397.
[5] R. Biddle, M. Mannan, P.C. van Oorschot, T. Whalen, User study, analysis, and usable security of passwords based on digital objects, IEEE Trans. Inform. Forensics Security 6 (2011) 970–979.
[6] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford Unifersity Press, New York, NY, USA, 1995.
[7] S. Bleha, C. Slivinsky, B. Hussien, Computer access security systems using keystroke dynamics, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 1217–1222.
[8] M. Burdon, Commercializing public sector information privacy and security concerns, IEEE Technol. Soc. Mag. 28 (2009) 34–40.
[9] P. Campisi, E. Maiorana, M.L. Bosco, A. Neri, User authentication using keystroke dynamics for cellular phone, IET Signal. Process. 3 (2009) 333–341.
[10] K. Challita, H. Farhat, K. Khaldi, Biometric authentication for intrusion detection systems, in: Proceedings of the First International Conference on Integrated Intelligent Computing (ICIIC 2010), Bangalore, India, pp. 195–199.

[11] T.Y. Chang, Dynamically generate a long-lived private key based on password keystroke features and neural network, Inform. Sci. 211 (2012) 36–47.

[12] W. Chen, W. Chang, Applying hidden markov models to keystroke pattern analysis for password verification, in: Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2004), Las Vages, NV, USA, pp. 467–474.

[13] N. Clarke, S. Furnell, Biometrics – the promise versus the practice, Comput. Fraud Security 2005 (2005) 12–16.

[14] N. Clarke, S. Furnell, B. Lines, Application of keystroke analysis to mobile text messaging, in: Proceedings of the Thrid Security Conference, Las Vagas, NV, USA.

[15] N. Clarke, S. Furnell, B. Lines, P. Reynolds, Subscriber authentication for mobile phones through the implementation of keystroke dynamics, in: Proceedings of the Thrid International Network Conference (INC 2002), Plymouth, pp. 347–355.

[16] J.F. Connolly, E. Granger, R. Sabourin, An adaptive classification system for video-based face recognition, Inform. Sci. 192 (2012) 50–70.

[17] H. Crawford, Keystroke dynamics: Characteristics and opportunities, in: Proceedings of the Eights Annual International Conference on Privacy, Security, and Trust (PST 2010), Ottawa, ON, Canada, pp. 205–212.

[18] H. Davoudi, E. Kabir, A new distance measure for free text keystroke authentication, in: Proceedings of the 14th International Computer Conference (CSICC 2009), Tehran, Iran, pp. 570–575.

[19] A. Dearle, Toward ubiquitous environments for mobile users, IEEE Internet Comput. 2 (1999) 22–32.

[20] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley & Sons, New York, NY, USA, 2001.

[21] C. Feher, Y. Elovici, R. Moskovitch, L. Rokach, A. Schclar, User identity verification via mouse dynamics, Inform. Sci. 201 (2012) 19–36.

[22] J. Filho, E. Freire, On the equalization of keystroke timing histograms, Pattern Recognit. 27 (2006) 1440–1446.

[23] R.A. Fisher, Questions and answers #14, Am. Stat. 2 (1948) 30–31.

[24] R. Giot, M. El-Abed, C. Rosenberger, Keystroke dynamics with low constraints SVM based passphrase enrollment, in: Proceedings of the Third International Conference on Biometric: Theory, Applications, and Systems (BTAS 2009), Washinton, DC, USA, pp. 1–6.

[25] D. Gunetti, C. Picardi, Keystroke analysis of free text, ACM Trans. Inform. Syst. Security 8 (2005) 312–347.

[26] S. Harmeling, G. Dornhege, D. Tax, F. Meinecke, K.R. Müller, From outliers to prototypes: ordering data, Neurocomputing 69 (2006) 1608–1618.

[27] A. Hiltgen, T. Kramp, T. Weigold, Secure internet banking authentication, IEEE Security Privacy 4 (2006) 21–29.

[28] D. Hosseinzadeh, S. Krishnan, Gaussian mixture modeling of keystroke patterns for biometric applications, IEEE Trans. Syst., Man, Cybernet. – Part C: Appl. Rev. 38 (2008) 816–826.

[29] J. Hu, D. Gingrich, A. Sentosa, A k-nearest neighbor approach for user authentication through biometric keystroke dyanmics, in: Proceedings of the IEEE International Conference on Communications (ICC 2008), New York City, NY, pp. 1556–1560.

[30] N.L. Johnson, S. Kotz, N. Balakrishnan, Continuous Univariate Distributions, vol. 2, John Wiley & Sons, New York, NY, USA, 1995.

[31] R. Joyce, G. Gupta, Identity authentication based on keystroke latencies, Commun. ACM 33 (1990) 168–176.

[32] Y. Kaneko, Y. Kinpara, Y. Shiomi, A Hamming distance-like filtering in keystroke dynamics, in: Proceedings of the Ninth Annual International Conference on Privacy, Security, and Trust, Montreal, QC, Canada, pp. 93–95.

[33] P. Kang, S. Cho, A hybrid novelty score for novelty detection and its use in keystroke dynamics-based user authentication, Pattern Recognit. 42 (2009) 3115–3127.

[34] S. Karatzouni, N. Clarke, Keystroke analysis for thumb-based keyboards on mobile devices, in: Proceedings of the 22nd IFIP International Information Security Conference (IFIP SEC 2007), Sandton, South Africa, pp. 235–263.

[35] M. Karnan, M. Akila, Personal authentication based on keystroke dynamics using soft computing techniques, in: Proceedings of the Second International Conference on Communication Software and Networks (ICCSN 2010), Singapore, pp. 334–338.

[36] M. Karnan, M. Akila, N. Krishnaraj, Biometric personal authentication using keystroke dynamics: a review, Appl. Soft Comput. 11 (2011) 1565–1573.

[37] K. Killourhy, R. Maxion, Comparing anomaly-detection algorithms for keystroke dynamics, in: Proceedings of the IEEE/IFIP International Conference on Dependable Systems & Networks (DSN 2009), Lisbon, Portugal, pp. 125–134.

[38] W. Lou, K. Ren, Security, privacy, and accountability in wireless access networks, IEEE Wireless Commun. Mag. 16 (2009) 80–87.

[39] L. Ma, T. Tan, Y. Wang, D. Zhang, Personal identification based on iris texture analysis, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003) 1519–1533.

[40] R.A. Marino, F.H. Alvarez, L.H. Encinas, A crypto-biometric scheme based on iris-templates with fuzzy extractors, Inform. Sci. 195 (2012) 91–102.

[41] G. Marsaglia, W.W. Tsang, J. Wang, Evaluating Kolmogorov's distribution, J. Stat. Softw. 8 (2003) 1–4.

[42] A. Messerman, T. Mustafić, S. Ahmet, Camtepe, S. Albayrak, Continuous and non-intrusive identity verification in real-time environments based on free-text keystroke dynamics, in: Proceedings of the International Joint Conference on Biometrics (IJCB 2011), Washington, DC, USA, pp. 1–8.

[43] F. Monrose, M.K. Reiter, S. Wetzel, Password hardening based on keystroke dynamics, Int. J. Inform. Security 1 (2002) 69–83.

[44] F. Monrose, A.D. Rubin, Keystroke dynamics as a biometric for authentication, Future Gener. Comput. Syst. 16 (2000) 351–359.

[45] B. Ngugi, B.K. Kahn, M. Tremaine, Typing biometrics: impact of human learning on performance quality, J. Data Inform. Qual. 2 (2011) 11:1–21.

[46] M. Obaidat, B. Sadoun, A simulation evaluation study of neural network techniques to computer user identification, Informa. Sci. 102 (1997) 239–258.

[47] A. Peacock, X. Ke, M. Wilkerson, Biometric recognition: security and privacy concerns, IEEE Security Privacy 2 (2004) 40–47.

[48] S. Prabhakar, S. Pankanti, A.K. Jain, Typing patterns: a key to user identification, IEEE Security Privacy 1 (2003) 33–42.

[49] Y. Sheng, V.V. Phoha, S.M. Rovnyak, A parallel decision tree-based method for user authentication based on keystroke patterns, IEEE Trans. Syst., Man, Cybernet. – Part B: Cybernet. 35 (2005) 826–833.

[50] T. Shimshon, R. Moskovitch, L. Rokach, Y. Elovici, Continuous verification using keystroke dynamics, in: Proceedings of the International Conference on Computational Intelligence and Security (CIS 2010), Nanning China, pp. 411–415.

[51] W. Shu, D. Zhang, Automated personal identification by palmprint, Opt. Eng. 37 (1998) 2659–2662.

[52] N.V. Smirnov, Tables for estimating the goodness of fit of empirical distributions, Ann. Math. Stat. 19 (1948) 279–281.

[53] D. Stefan, X. Shu, D. Yao, Robustness of keystroke-dynamics based biometrics against synthetic forgeries, Comput. Security 31 (2012) 109–121.

[54] C.C. Tappert, S.H. Cha, M. Villani, R.S. Zack, A keystroke biometric system for long-text input, Int. J. Inform. Security Privacy 4 (2010) 32–60.

[55] C.C. Tappert, M. Villani, S.H. Cha, Keystroke biometric identification and authentication on long-text input, in: L. Wang, X. Geng (Eds.), Behavioral Biometrics for Human Identification: Intelligent Applications, Idea Group Inc., 2010, pp. 342–367.

[56] D. Tax, R. Duin, Support vector domain description, Pattern Recognit. Lett. 20 (1999) 1191–1199.

[57] P.S. Teh, A.B.J. Teoh, C. Tee, T.S. Ong, A multiple layer fusion approach on keystroke dynamics, Pattern Anal. Appl. 14 (2011) 23–36.

[58] C. Tompson, Next-generation virtual worlds: architecture, status, and directions, IEEE Internet Comput. 15 (2011) 60–65.

[59] A.A. Toptsis, J. Majonis, PAAKL: Password authentication with behavioral metrics, in: Proceedings of the 34th IEEE Annual Computer Software and Applications Conference (COMPAC 2010), Seoul, South Korea, pp. 351–356.

[60] Y. Uzun, K. Bicakci, A second look at the performance of neural networks for keystroke dynamics using a publicly available dataset, Comput. Security 31 (2012) 717–726.

[61] M. Villani, C. Tapert, G. Ngo, J. Simone, H.S. Fort, S.H. Cha, Keystroke biometric recognition studies on long-text input under ideal and application-oriented conditions, in: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW 2006), New York City, NY, pp. 39–47.

[62] D. Voth, Face recognition technology, IEEE Intell. Syst. 18 (2003) 4–7.

[63] B.L. Welch, The generalization of "Student's" problem when several different population variances are involved, Biometrika 34 (1947) 28–35.

[64] N. Yager, T. Dunstone, The biometric menagerie, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 220–230.

[65] J. Yan, A. Blackwell, R. Anderson, A. Grant, Password memorability and security: empirical results, IEEE Security Privacy 2 (2004) 25–31.

[66] R.S. Zack, C.C. Tappert, S.H. Cha, Performance of a long-text-input keystroke biometric authentication system using an improved k-nearest-neighbor classification method, in: Proceedings of the Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS 2010), Washington, DC, pp. 1–6.

[67] Y. Zhang, G. Chang, L. Liu, J. Jia, Authenticating user's keystroke based on statistical models, in: Proceedings of the Fourth International Conference on Genetic and Evolutionary Computing (ICGEC 2010), Shenzhen, China, pp. 578–581.

[68] Y. Zhong, Y. Deng, A. Jain, Keystroke dynamics for user authentication, in: Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, RI, pp. 117–123.