Accurate Continuous and Non-intrusive User Authentication with Multivariate Keystroke Streaming

Abdullah Alshehri, Frans Coenen and Danushka Bollegala
Department of Computer Science, University of Liverpool, Liverpool, United Kingdom
{a.a.alshehri, coenen, danushka.bollegala}@liverpool.ac.uk

Keywords: Keystroke Time Series, Continuous Authentication, Keystroke Streams, Behavioral Biometric

Abstract: In this paper, we demonstrate a novel mechanism for continuous authentication of computer users using keystroke dynamics. The mechanism models keystroke timing features, Flight time (the time between consecutive keys) and Hold time (the duration of a key press), as a multivariate time series which serves to dynamically capture typing patterns in real/continuous time. The proposed method differs from previous approaches for continuous authentication using keystroke dynamics, founded on feature vector representations, which limited real-time analysis due to the computationally expensive processing of the vectors, and which also yielded poor authentication accuracy. The proposed mechanism is compared to a feature vector based approach, taken from the literature, over two datasets. The results indicate superior performance of the proposed multivariate time series mechanisms for continuous authentication using keystroke dynamics.

1 INTRODUCTION

Distance learning (eLearning) and Massive Open Online Courses (MOOCs) have witnessed a rapid growth over the last decade (Clark and Mayer, 2016). Consequently, an increasing number of people are taking online assessments and exams remotely. As a result, user authentication has become an issue; is the person taking the assessment the person who they say they are? Traditional one-off validation, such as the utilisation of passwords and usernames (also known as static authentication (Bours, 2012)) is clearly not sufficient. The challenge is not just how to ensure that the person is who they say they are at the beginning of the assessment but throughout the assessment. What is required is real time continuous authentication rather than static authentication.

Keystroke dynamics (typing patterns) are a promising biometric recognition mechanism that can provide the desired continuous authentication (Bours, 2012). They offer the advantage that no special equipment is required such as in the case of continuous iris or fingerprint recognition. The idea is motivated by the observation that keyboard behaviour (typing rhythm and style) varies between individuals (Gaines et al., 1980). Keyboard behaviour can be expressed in the form of patterns made up of keystroke timing attribute-values (keystroke dynamics) describing the timing information associated with key-hold times ($\mathcal{KH}^t$) and flight times ($\mathcal{F}^t$) (Gaines et al., 1980). The first is the duration of holding down a key; the second is the time from the first key press to the last key release of *n*-graphs. An *n*-graph in this context is a sequence of keyboard characters; if we have one character this is called a *monograph*, two characters a *digraph*, three characters a *trigraph*, and so on (Gunetti and Ruffo, 1999; Ahmed and Traore, 2014).

Much previous work directed at user (keyboard) authentication using keystroke dynamics has focussed on Keystroke Static Authentication (KSA) where typing patterns are extracted from short predefined texts, for example, passwords or pin numbers, to strengthen user credentials (Joyce and Gupta, 1990; Killourhy and Maxion, 2009; Syed et al., 2014). The authentication process utilises keystroke dynamics to create a feature vector (often referred to as a *typing profile*), for example, made up of the mean and standard deviation of digraph flight times. Then, given a previously unseen profile, allegedly belonging to a certain user, this can be authenticated by comparing it to a stored profile for the indicated user, by computing the similarity between the two feature vectors. The idea of using keystroke dynamic feature vectors for free text, Keystroke Continuous Authentication (KCA), has also been proposed (Shepherd, 1995; Dowland and Furnell, 2004; Gunetti and Picardi, 2005; Ahmed and Traore, 2014). This is typically conducted in the context of specific *n*-graphs (such as “th”, “ing”, “tion” and so on). The generation of the desired feature vector (profile), and the consequent authentication, is typically carried out once the subject has finished typing.

The feature vector representation when used in the context of KCA, as described above, has a number of disadvantages (Ahmed and Traore, 2014). The most significant is the resource required to generate the feature vectors, which is why authentication is typically conducted on typing completion rather than in real time as would be desirable in the context of online assessment. The reason for this is that a great many *n*-graphs need to be considered so as to acquire an effective typing profile. By increasing the number of *n*-graphs to, for example, all digraphs, real time KCA (while the subject is typing) becomes intractable given the size of the feature vectors that need to be generated and compared.

In this paper, a new approach to KCA is proposed that takes into consideration all keystroke features, rather than those associated with specific digraphs, by considering typing behaviour in terms of a multivariate time (keystroke) series. More formally, each keystroke is considered to be a multivariate discrete (indexed) temporal event (p_i) forming a sequence of multidimensional events $\{p_1, p_2, \dots\}$ where each p represents the timing information of flight $\mathcal{F}^t$ and key-holds $\mathcal{KH}^t$ timings. The idea is to conduct real time authentication by continuously extracting subsequences of a keyboard usage multivariate time series, that are representative of typing behaviour and which can consequently be used for real time KCA. In the proposed method, the subsequences are extracted using a non-overlapping sliding window (of length ω). The real time authentication is conducted by comparing the most recent subsequence with the previous subsequence extracted during the typing session. Of course, on start up, the subject’s identity needs to be initially confirmed in a “traditional” manner with reference to a stored typing pattern. Time series similarity was calculated using Dynamic Time Warping (DTW) (Berndt and Clifford, 1994) which it results in a *warping distance* which can, in turn, be used as a similarity measure.

The main contributions of the work presented in this paper are as follows. A process for real time KCA founded on the concept of multivariate time series, as opposed to the feature vector-based approaches proposed to date, that is independent of keyboard layout. The mechanism is also privacy preserving in that knowledge of which keys are actually being pressed is not required. The proposed mechanism, although intended for use with respect to online assessment, has general applicability. For example, it may equally well be used to detect certain human conditions, such as Parkinson’s disease, as described in (Giancardo et al., 2016) or to detect keyboard user emotions as described in (Raja and Sigg, 2016). A further contribution is the novel manner in which typing behaviour is captured in the form of a multivariate time series representation. Note also that in the proposed mechanism, other than on start up, there is no requirement to compare to a “bank”[1] of user profiles for the claimed user.

The rest of this paper is organised as follows. Section 2 reviews the background and the related work concerning KCA feature representation. Section 3 presents some definitions, and some preliminaries, concerning the keystroke multivariate time series representation. The proposed KCA process is then introduced in Section 4. The evaluation of the proposed approach is reported on in Section 5. Finally, the work is concluded in Section 6.

2 PREVIOUS WORK

As noted in Section 1, most previous work directed at KCA, although limited, has been founded on the use of statistical measurements to define features from predefined sets of *n*-graphs (see for example (Dowland and Furnell, 2004; Gunetti and Picardi, 2005; Ahmed and Traore, 2014)).

One of the earliest examples of this approach can be found in (Dowland and Furnell, 2004) where the focus was on the most frequent occurring digraphs. In this case, the created typing profiles comprised the mean and standard deviation of the flight time for different digraphs. These profiles could then be used for authentication purposes. The limitation of this mechanism was that to obtain an accurate performance, a substantial number of digraphs (for each user) was required. The authors stated that, in their experiments, an average of 6,390 digraphs were required to construct a reliable typing profile. In (Gunetti and Picardi, 2005) the average flight time for shared digraphs and trigraphs (between two typing samples) was used to create typing profiles that were stored in arrays. Given a new profile, this was compared to existing profiles by comparing the array ordering of the new profile with existing profiles using the R measure[2]. Thus, authenticating a new sample required comparison with all stored sample profiles

[1] A repository holding a collection of relevant typing profiles.

[2] An idea inspired by Spearman’s rank correlation coefficient.

(reference profiles), a computationally expensive process. In the reported evaluation, 600 reference profiles were considered (generated from 40 users, each with 15 samples); the time taken for a single match, in this case, was 140 seconds (using a Pentium IV, 2.5 GHz). However, for most KCA applications, the current sample need only be compared against the claimed user's reference profile.

An issue with the work presented in both (Dowland and Furnell, 2004) and (Gunetti and Picardi, 2005) was the size of feature vectors used to represent typing profiles which required a substantial resource to collect, hence authentication was conducted on typing completion. To address this disadvantage, the idea presented in (Ahmed and Traore, 2014) was to collect only a small number of features (average hold time for monograms and average flight time for frequently occurring digraphs) and predict the values for missing features to complete a new typing profile. Once completed this could be compared with the appropriate reference profiles (the stored profiles for the person the new user claimed to be). A neural network based classifier was trained to predict the missing values. The proposed mechanism worked well under controlled experimental conditions, although ideally, we would like to undertake the evaluation in an uncontrolled setting (as in the case of (Dowland and Furnell, 2004) and (Gunetti and Picardi, 2005)).

There has been very little previous work directed at time series analysis in the context of user authentication using keyboard dynamics. In (Richardson et al., 2014) a streaming algorithm was proposed for which a potential suggested application domain was real time KCA, although no further investigation was ever conducted. In (Alshehri et al., 2016a) and (Alshehri et al., 2016b) the concept of using single-variate time series with respect to keystroke dynamics was considered, but only in the context of static authentication. Therefore, the work presented in this paper is directed at continuous authentication using multivariate time series. The first hypothesis investigated in this paper is the idea that when incorporating more than one keystroke timing feature, in a keyboard time series representation, the quality of the user authentication can be improved significantly. The second hypothesis is that the computational complexity of the proposed mechanism, although clearly exceeding the complexity associated with single-variate analysis, will be such that real time user authentication will be viable. To the best knowledge of the authors, real time KCA using multivariate time series has not been considered previously in the literature.

3 KEYSTROKE TIME SERIES PRELIMINARIES

The generic concept of time series is well described in the literature (see for example (Wang et al., 2013)). This section presents the application of the concept to keystroke multivariate time series starting with a formal description of what a keystroke time series is.

Definition 1: *A keystroke time series $\mathcal{K}_{ts}$ is an ordering of keyboard events $\{p_1, p_2, \dots, p_n\}$ where $n \in \mathbb{N}$ is the length of the series.*

Definition 2: *A dimensional keyboard event (keystroke) $p_i \in \mathcal{K}_{ts}$ is parametrised as a tuple of the form $\langle t_i, k_i \rangle$, where t_i is an identifying index and k_i is a collection of multivariate keystroke timing features.*

The keystroke timing features used in the proposed representation are flight time $\mathcal{F}^t$ and and key-hold $\mathcal{KH}^t$. That is, each event $p_i \in \mathcal{K}_{ts}$ can be given as:

$$p_i \rightarrow \langle t_i, k_i \rangle \mid t = [0, n), k = \{\mathcal{F}^t, \mathcal{KH}^t\} \quad (1)$$

such that a keystroke time series can be formulated as a multivariate series of the form $\{\langle t_1, \mathcal{F}_1^t, \mathcal{KH}_1^t \rangle, \langle t_2, \mathcal{F}_2^t, \mathcal{KH}_2^t \rangle, \dots\}$.

The fundamental idea is then to extract short keystroke time series subsequence, using a moving window of size ω, and use these for real time KCA by comparing the current subsequence with earlier subsequences so as to confirm that the subject remains who they say they are and has not been replaced by another subject. For the implementation presented later in this paper, Dynamic Time Warping (DTW), as described in (Berndt and Clifford, 1994), was used to determine the similarity between multivariate subsequences. DTW was used because, unlike other similarity devices such as Euclidean distance, it has the ability to capture shape offsets (Lines et al., 2012). In other words, it supports non-linear similarity determination. For completeness the process of DTW is briefly described in Sub-section 3.1 below.

3.1 Multivariate Similarity Checking Using DTW

The concept of multivariate time series, although not new (Vlachos et al., 2003), has attracted much recent attention in the literature (Hu et al., 2013; Cao and Liu, 2016). There have been many approaches proposed to determine the similarity between multivariate time series; in this paper, because we are interested in the effectiveness of the multivariate keystroke time series representation in the context of KCA, a multivariate time series is considered in terms of its component single time series. Thus in our case similarity is expressed in terms of two *warping path distances*, determined using the technique presented (Vlachos et al., 2003), from which an average distance is obtained.

Thus, given two multivariate keystroke time series subsequences:

$$\mathcal{K}_{ts1} = \{(a_1^{\{x_1,y_1\}}), (a_2^{\{x_2,y_2\}}), \ldots, (a_\omega^{\{x_w y_w\}}\}$$

and

$$\mathcal{K}_{ts2} = \{(b_1^{\{x_1,y_1\}}), (b_2^{\{x_2,y_2\}}), \ldots, (b_\omega^{\{x_w y_w\}}\}$$

where ω is the subsequence (window) length, and the set $\{x,y\}$ is the coordinate set, $y \to \mathcal{F}^t$ or $y \to \mathcal{KH}^t$. The DTW commences with the creation of a matrix $\boldsymbol{X}$ of size $\omega \times \omega$ holding the absolute Euclidean distances d between each point $a_i \in \mathcal{K}_{ts1}$ and all points $b_i \in \mathcal{K}_{ts2}$, such that:

$$d(a_i,b_i) = \sqrt{\left((a_i^{(x_i)} - b_i^{(x_i)}) + (a_i^{(y_i)} - b_i^{(y_i)})\right)^2} \quad (2)$$

The *warping path* is then the path from location $\langle 0,0 \rangle$ to location $\langle \omega,\omega \rangle$ in the matrix $\boldsymbol{X}$, that features the lowest Euclidean distance values in the matrix. The *warping path distance* (w) associated with this path is the sum of the individual Euclidean distances normalised by the distance associated with the path that features the maximum Euclidean distance values in the matrix $\boldsymbol{X}$. If $w = 0$ the subsequences $\mathcal{K}_{ts1}$ and $\mathcal{K}_{ts2}$ are identical, otherwise they are different by some degree. Thus w is a measure of similarity. The principle of DTW is illustrated in Figure 1. The Figure shows the optimal warping path with respect to two scenarios: Figure 1(a) shows an example of the warping path that results when comparing two keystroke time series from the same user, but typing different texts; whereas Figure 1(b) shows an example of the warping path that results when comparing two keystroke time series for different users typing the same text. In both cases $\omega = 100$. From the figures, it can be noted that the warping path is shorter with respect to the same user than in the different user case.

To decide whether two subsequences can be considered to be similar or not, a threshold value σ was employed. In the context of the proposed real time KCA, the value of σ was derived dynamically; how this is achieved is discussed in Sub-section 4.2 below. Note that the length of both keystroke time series ($\mathcal{K}_{ts1}$ and $\mathcal{K}_{ts2}$) does not have to be equal, the reason they are equal in the above explanation is simply because this was the case with respect to the proposed real time KCA approach which uses a fixed window size ω.

4 CONTINUOUS/REAL TIME KCA

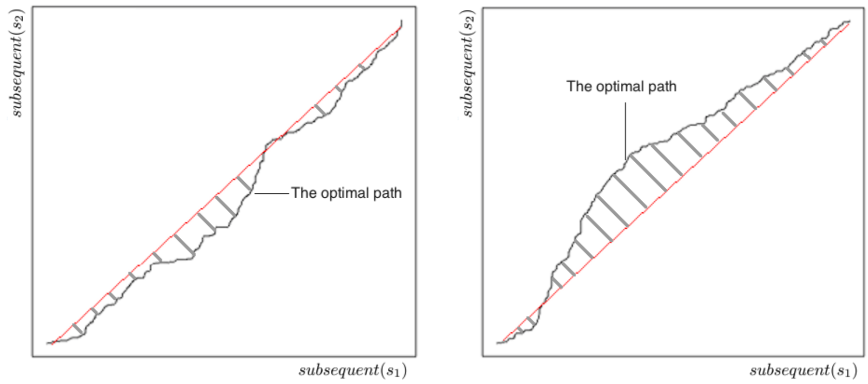
The proposed real time KCA process is presented in this section. As established above, the process comprises the comparison of time series subsequences collected, using a sliding window of size ω, whilst the subject is typing. As already noted the keystroke dynamics used were flight time ($\mathcal{F}^t$) and key-hold time ($\mathcal{KH}^t$). An issue with fight time is that the value can be large, for example when the subject pauses during their typing or as a result of an "away from keyboard" event. For each collected time series it was thus necessary to address this issue, before commencing any further KCA authentication. How this was achieved is presented in Sub-section 4.1. Whatever the case, on "start-up", it was first necessary to authenticate the user with respect to a set of stored profiles for the claimed user. At the same time, we need to establish a value for σ. How this is achieved is presented in Sub-section 4.2. Once we have authenticated the user we can commence the monitoring process as described in Sub-section 4.3.

4.1 Noise Reduction for Flight Time

In the foregoing, it was noted that a given flight time value $\mathcal{F}^t$ might be greater than normal because the subject has paused during his/her typing. Essentially such high values introduce noise into the real time KCA process. To address this issue, a limit was placed on the $\mathcal{F}^t$ values in a given time series $\mathcal{K}_{ts}$ using a second threshold value φ. In other words, given a specific $\mathcal{F}^t$ value in access of φ, the value was reduced to φ. In the evaluation presented later in this paper a range of values for φ were considered, ranging from 0.75 to 2.00 seconds, increasing in steps of 0.25 seconds ($\{0.75, 1.00, 1.25, 1.50, 1.75, 2.00\}$).

4.2 User Authentication on Start-up

At the beginning of the process, it will be first necessary to confirm that the user is who(s)he says (s)he is. This is done by comparing the first extracted subsequence, $\mathcal{S}_i$, with a collection of *reference subsequences* (of the same length ω) extracted from a sample typing profile $\mathcal{K}_{ts}$, known to belong to the claimed subject and obtained previously. Note that in this case, $\mathcal{K}_{ts}$ needs to be substantially greater than the maximum anticipated value for ω, so that



(a) (b)

Figure 1: The application of DTW: (a) warping path for the same user typing different texts, and (b) warping path for different users typing the same text.

a number of subsequences can be extracted. $\mathcal{S}_1$ is then compared, using DTW, with the reference subsequences set, and an average warping path distance $\bar{w}$ obtained. The required value for σ is obtained by comparing all the profile subsequences pair combinations in $\mathcal{K}_{ts}$ and obtaining a set of warping path values $W = \{w_1, w_2, \dots\}$. The average value of W is then calculated and used as the value of σ which is compared with $\bar{w}$ to determine whether the user is who they say they are. Note that the σ value derived in this manner is used throughout the rest of the real time KCA process for the current typing session. Note also that the window size ω, is user defined. For the experiments reported on later in this paper a range of ω values were considered from 25 to 150 increasing in steps of 25 ($\{25, 50, 75, 100, 125, 150\}$).

4.3 Continuous User Authentication

Once the initial authentication had been completed, and a value for σ obtained, the real time KCA can be commenced. The pseudo code for the process is presented in Algorithm 1. The inputs are: (i) the desired window size ω and (ii) the similarity threshold σ (derived as described above) and the desired φ limit value for $\mathcal{F}^t$. The process loops continuously until the keyboard session is terminated (the subject completes the assessment, times out or logs-out); on each iteration, the flight time $\mathcal{F}^t$ and $\mathcal{KH}^t$ values are simultaneously recorded. Each $\mathcal{F}^t$ value is checked to ensure that it is within the limit (φ). This is done using the function *checkLimit* (line 8) which returns the $\mathcal{F}^t$ value if it is less than φ, and φ otherwise (as described above). $\mathcal{F}^t$ and $\mathcal{KH}^t$ are then appended to the keystroke time series $\mathcal{K}_{ts}$ so far. Note that $\mathcal{K}_{ts}$ is built up as the process progresses. The *counter* is monitored and subsequences are extracted whenever the number of keystrokes reaches ω. For the first collected subsequence ($\mathcal{S}_1 \in \mathcal{K}_{ts}$) this is the start-up situation and consequently $\mathcal{S}_1$ is processed as described above in Sub-section 4.2 and a report returned. Otherwise the subsequence $\mathcal{S}_i$ is compared to the previous subsequence $\mathcal{S}_{i-1}$ ($i > 1$), using the DTW process as described in Sub-section 3.1.

Algorithm 1 Multivariate KCA process

Input: ω, σ, φ.
Output: Authentication commentary.

1: $counter = 0$
2: $\mathcal{K}_{ts} = \emptyset$
3: **loop**
4:     **if** end of session signal received **then**
5:         break
6:     **end if**
7:     $\mathcal{F}^t$ = current keystroke dynamic (flight time)
8:     $\mathcal{F}^t = checkLimit(\mathcal{F}^t, \varphi)$ ▷ Noise reduction.
9:     $\mathcal{K}_{ts} = \mathcal{K}_{ts} \cup \mathcal{F}^t$
10:     $counter{+}{+}$
11:     **if** $REM(counter/\omega) == 0$ **then**
12:         $\mathcal{S}_i$ = subsequence $\{\mathcal{K}_{ts_{counter-\omega}} \dots \mathcal{K}_{ts_{counter}}\}$
13:         **if** $counter = \omega$ **then**
14:             Start-up: authenticate $\mathcal{S}_i$ w.r.t reference profiles and σ
15:         **else**
16:             Authenticate $\mathcal{S}_i$ w.r.t. $\mathcal{S}_{i-1}$ and σ
17:         **end if**
18:     **end if**
19: **end loop**

5 EVALUATION

The evaluation of the proposed multivariate time series-based approach to continuous/real time keystroke authentication is presented in this section. For the evaluation two data sets were used; these are thus first discussed in Sub-section 5.1. Experiments were conducted to: (i) evaluate the processing time required to generate user profiles, (ii) determine how well the proposed approach performed in terms of the detection of impersonators, (iii) compare the operation of the proposed multivariate keystroke time series with the use of univariate keystroke time series for KCA (when we only use the $\mathcal{F}^t$ feature as proposed in the previous study presented in (Alshehri et al., 2016b)) and (iv) compare the operation of the proposed approach with the traditional feature vector representation based approach when applied to KCA. The results obtained with respect to these objectives are discussed in further detail in Sub-sections 5.2, 5.3, 5.4 and 5.5 respectively. The metrics used for the evaluation were: (i) authentication accuracy (Acc.), (ii) the False Acceptance Rate (FAR), (iii) the False Rejection Rate (FRR) and (iv) Runtime (seconds). Note that FAR and FRR are the traditional metrics used to measure the performance of biometric authentication systems (Polemi, 1997). Two-fold cross validation was conducted, hence results presented below are average results from two cross validations.

5.1 Datasets and Experimental Setting

The experiments reported on in this section were conducted using two datasets: (i) ACB, collected by the authors, and (ii) VHHS obtained from (Vural et al., 2014). Note that each dataset comprised keystroke dynamics from free text sessions obtained from volunteer keyboard users, however each featured differing characteristics, thus different: (i) number of subjects, (ii) lengths of typed samples, (iii) subject matter (specified or unspecified) and (iv) environments in which the samples were collected (laboratory or otherwise). Each is discussed in some further detail below. Table 1 provides some statistics concerning both datasets.

For the evaluation, the record associated with each subject was split into two so that one-half could be used as the reference profile and the other to simulate a typing stream. Recall that for the proposed real time KCA, the reference profiles were used to: (i) derive a value for σ and (ii) for "start-up authentication".

5.1.1 ACB Dataset

The ACB dataset was collected anonymously using undergraduates, postgraduates and staff at the authors' university; the total number of subjects participating was 30. Some additional information regarding the gender and/or age were deliberately ignored for two reasons: (i) to minimise the resource required by subjects providing the data, and (ii) so as to focus only on comparing user typing patterns for user authentication, not on drawing any conclusions about the nature of keyboard usage behaviour in the context of (say) age or gender. Furthermore, the subjects were asked to provide answers to general questions to simulate the way that online assessments might be run. Thus, the subjects were allowed to undertake the exercise in their own time using whatever keyboard, operating system and browser they had at hand. The website interface from where the ACB dataset was collected can be found at: (`http://cgi.csc.liv.ac.uk/hsaalshe/WBKTR3.html`).

5.1.2 VHHS Dataset

The VHHS dataset was generated by the authors of (Vural et al., 2014) and comprised both fixed and free text samples. For the evaluation reported here, only the free text samples were considered so that meaningful comparisons could be made with the results obtained using the ACB dataset. Note also that the VHHS dataset was collected under laboratory conditions, not the case with respect to the ACB dataset. A total of 39 subjects were recruited.

5.2 Typing Profile Generation

Table 2 shows the run-time values (seconds) taken to generate the reference profiles, required on start-up of the proposed real time KCA process, using different ω values. The "per subject" values were obtained by dividing the total run time with the number of records in the dataset from Table 1. The experiment was run twice, each time with a different half of the data (two-fold cross-validation), the results presented in the table are thus average values. From the table, it can be clearly seen that, regardless of the dataset, processing time increased with ω. This was to be expected because the computation time required by the DTW would increase as the size of the subsequence considered increased (even though there might be less of them). There are well known solutions in the literature to mitigate against the complexity of DTW (Itakura, 1975; Sakoe and Chiba, 1978), no such mitigation was applied with respect to the experiments

Table 1: Summary of Evaluation Datasets.

Dataset	# Subject.	Environment.	Language used.	Average size	Standard Deviation (SD)
ACB	30	Free	English	4625	1207
VHHS	39	Lab.	English	4853	1021

Table 2: The time taken (in seconds) to construct subject reference profiles.

ω	Entire Dataset		Per Subject	
	ACB	VHHS	ACB	VHHS
25	1.007	1.180	0.034	0.030
50	1.969	1.570	0.066	0.040
75	2.980	3.523	0.099	0.090
100	5.095	6.244	0.170	0.160
125	7.940	9.742	0.265	0.250
150	12.000	14.130	0.400	0.362

reported here. Nevertheless, we obtained much better efficiency compared with the feature vector approach as stated in (Gunetti and Picardi, 2005) where the time taken to construct typing profile was 140 seconds. Thus, we have a worst case runtime ratio 1 : 23; using $\omega = 100$ (the best performing value for ω as demonstrated later in this section) we have a ratio of 1 : 29, which is a significant "speed-up".

5.3 User KCA Performance

For each dataset, the continuous typing process was simulated by presenting the keystroke dynamics for each subject in the form of a data stream. In each case, the data stream was appended with a randomly selected second data stream from another user. The idea being to simulate one subject being impersonated by another half way through a typing session. For every comparison of a subsequence $\mathcal{S}_i$ with a subsequence $\mathcal{S}_{i-1}$ (line 17 in Algorithm 1) we recorded whether this was a True Positive (TP), False Positive (FP), False Negative (FN) or True Negative (TN). In this manner a *confusion matrix* was built up from which accuracy (Acc.), FAR and FRR could be calculated (using Equations 3, 4 and 5).

$$Acc = \frac{TP+TN}{TP+FP+FN+TN} \quad (3)$$

$$FAR = \frac{FP}{FP+TN} \quad (4)$$

$$FRR = \frac{FN}{FN+TP} \quad (5)$$

Table 3 presents the accuracy results obtained, with respect to both datasets, using $\omega = 25, 50, 75, 100, 125, 125$ and $\varphi = 0.750, 1.00, 1.25, 1.50, 1.75, 2.00$. Recall that in each case, the experiment was run twice each time with a different half of the data; the results presented are therefore averages. From the table, it can be observed that ω and φ values of 100 and 1.50 respectively tended to produce best results for all datasets (the best accuracy result recorded for ACB was 98.39%, whereas the best result for VHHS was 97.32%). Note also that the selection of φ does not seem to have as much impact as the selection of ω, and that the performance tends to decrease if the selected ω value is too big or too small.

Table 4 shows the best results obtained in terms of FAR and FRR (to give a clear comparison, the best accuracy values from Table 3, written in bold font, have been included in the table). The average recorded results for FAR and FRR are 0.036% and 1.971% respectively; an indicator that the proposed method gives promising results, and furthermore that it can be usefully employed as a behavioural biometric based approach for continuous user authentication.

5.4 Comparison with Univariate Keystroke Time Series

This subsection presents evaluation results obtained when the operation of the proposed mechanism was compared with univariate time series keystroke representation as introduced in (Alshehri et al., 2016b). More specifically univariate time series used for the experimentation were generated using the $\mathcal{F}^t$ keystroke dynamic. The results are presented in Table 5. From the table, it can be observed clearly that the univariate representation produced worse results, in term of accuracy, than the proposed multivariate approach. Best recorded results for ACB were 96.48% (with $\omega = 100$ and $\varphi = 1.25$), and for VHHS it was 96.54% (with $\omega = 100$ and $\varphi = 1.00$) compared with 98.39% and 97.32% respectively when using the multivariate approach (see Table 3). However, the run time results (not shown) indicate that the efficiency of constructing typing profiles during the start-up stage was slightly increased when using univariate time series compared with the use of multivariate time series. This was to be expected because the computational expensive of applying DTW univariate time series less than when applied to multivariate time series.

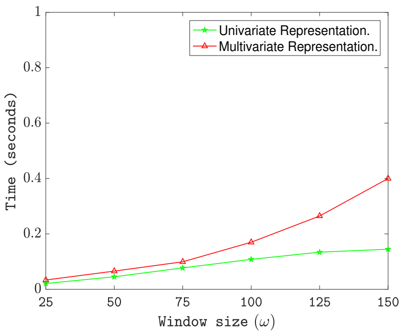
Figure 2 shows a comparison of run time versus ω. Inspection of the figure indicates that, as to be

Table 3: Accuracy recorded for Keystroke Multivariate Time Series realtime KCA (best results in bold font).

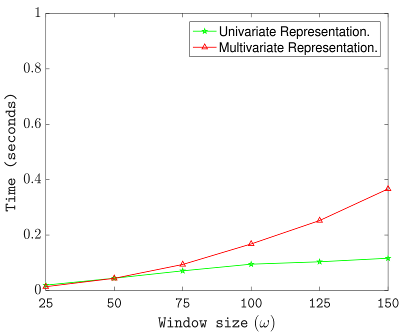
	ACB						VHHS					
ω \ φ	0.750	1.00	1.25	1.50	1.75	2.00	0.750	1.00	1.25	1.50	1.75	2.00
25	83.21	88.34	89.27	96.13	94.79	91.77	91.28	93.03	94.59	95.25	92.67	93.52
50	87.81	87.93	90.87	97.61	96.39	92.37	91.40	90.64	95.20	92.87	94.28	91.13
75	86.81	87.94	92.87	96.40	98.17	93.36	90.96	92.71	94.27	97.15	94.35	93.20
100	92.25	91.37	93.31	**98.39**	94.83	90.81	93.45	95.09	96.65	**97.32**	95.73	95.11
125	92.42	92.55	94.48	98.34	96.10	96.98	92.02	93.77	95.33	95.99	95.41	94.26
150	94.46	96.59	95.12	97.40	97.57	95.13	92.44	94.19	95.75	96.42	95.83	94.68

Table 4: Best Average Acc., FAR and FRR values obtained using the proposed real time KCA method ($\omega = 100$ and $\varphi = 1.50$).

Dataset	FAR (%)	FRR (%)	Accuracy (%)
ACB	0.045	1.922	**98.39**
VHHS	0.027	1.950	**97.32**
Average	0.036	1.936	**97.86**



(a)



(b)

Figure 2: Efficiency performance comparison for keystroke univariate time series and keystroke multivariate time series: (a) ACB dataset, (b) VHHS dataset.

expected, the use of univariate time series is more efficient than the use of multivariate time series. In conclusion, it can thus be observed that the multivariate representation gives better accuracy results than the univariate representation, but at a small efficiency expense. In other words, incorporating more keystroke dynamics produces a better KCA performance. Note that the limitation of the multivariate time series representation for KCA is a topic for future work.

5.5 Comparison with Feature Vector Approach

This subsection reports on the results obtained when the operation of the proposed real time KCA method is compared with the Feature Vector Representation (FVR) approach frequently encounter in previous work. As noted in Section 2, there are a number of reports where the feature vector approach has been applied to KCA although not in real time; the authentication is conducted on typing completion. Of these, the approach described in (Gunetti and Picardi, 2005) was selected for the comparison reported here because: (i) the study obtained, to the best knowledge of the authors, the best FAR and FRR results to date; and (ii) the approach was well explained in the literature, therefore it was easy to reproduce. The data for each subject, as before, was divided into two. One-half to be used to create a typing profile, and the other half as the test data. The features used were the $\mathcal{F}^t$ values for all shared digraph, trigraphs and quadgraphs in each corresponding sample. A typing profile was thus constructed for each user and comparisons conducted as described in (Gunetti and Picardi, 2005) (again see also Section 2). The results are presented in Table 6. For a comparison purpose, the table includes the accuracy (Acc.), FAR and FRR results presented earlier with respect to the proposed multivariate approach and comparator univariate approach. From the Table, it can be observed that the proposed multivariate time series-based approach to KCA obtained a better performance than the other two methods considered with respect to both datasets. Thus, confirming the hypothesis posed in the introduction to this paper that a multivariate time series representation would serve to better encapsulate keystroke dynamics than the univariate and feature vector based

Table 5: Accuracy recorded for Keystroke Univariate Time Series.

	ACB						VHHS					
ω \ φ	0.750	1.00	1.25	1.50	1.75	2.00	0.750	1.00	1.25	1.50	1.75	2.00
25	72.14	75.35	80.14	85.65	85.01	86.24	80.58	79.76	82.96	83.20	82.00	81.92
50	86.00	83.20	88.64	80.00	80.54	81.00	80.13	82.14	85.46	86.14	85.00	84.00
75	92.12	94.78	95.19	92.00	93.37	94.00	83.33	83.33	83.33	83.33	83.33	83.33
100	93.24	96.20	**96.48**	94.79	96.10	95.24	95.61	**96.54**	96.29	95.35	93.99	94.88
125	95.10	94.00	96.12	95.79	95.81	95.57	94.35	96.34	94.73	95.39	94.28	93.49
150	96.14	94.35	95.37	95.62	96.13	95.12	93.33	95.31	93.67	94.98	94.87	94.96

Table 6: Comparison of KCA using: (i) Multivariate Time Series, (ii) Univariate Time Series and (iii) a Feature Vector based representation (best results in bold font).

Dataset	Multivariate Time Series Based KCA			Univariate Time Series Based KCA			Feature Vector-Based KCA		
	FAR	FRR	Acc.	FAR	FRR	Acc.	FAR	FRR	Acc.
ACB	0.045	1.922	98.39	0.05	1.961	96.20	8.60	11.10	80.82
VHHS	0.027	1.950	97.32	0.03	1.970	94.83	9.09	7.13	89.15
Average	**0.036**	**1.936**	**97.86**	0.04	1.965	95.51	8.75	5.93	83.71

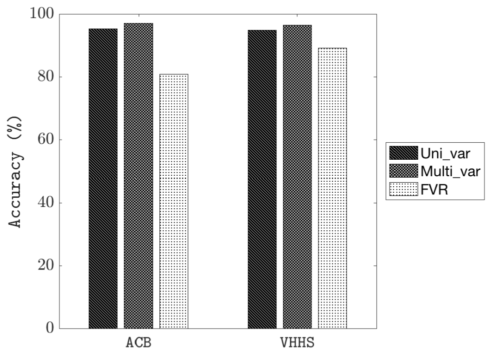


Figure 3: The accuracy obtained for multivariate time series representation, univariate time series representation and feature vector representation with respect to KCA application.

approach that has typically been used to date. For completeness Figure 3 presents the best accuracy results in graphical form.

6 CONCLUSION

In this paper, a novel mechanism for real time Keystroke Continuous Authentication (KCA) has been presented. The idea is to use sequences of keystroke dynamics in the form of multivariate time series, incorporating flight time $\mathcal{F}^t$ and key-hold time $\mathcal{KH}^t$, and to monitor such multivariate time series for the purpose of real time KCA. More specifically to periodically extract, from this data stream, subsequences that can be used for authentication purposes. In the proposed process, on start-up, the first subsequence (window) extracted for the subject will be compared to a reference profile (time series); subsequence $\mathcal{S}_i$ will then be compared to the immediate predecessor subsequence $\mathcal{S}_{i-1}$, and so on. In this manner continuous, real time, user authentication can take place. The advantage offered is first that the time series approach is much more efficient than the feature vector based approach used in earlier work. The presented evaluation indicated a speed up of approximately 1 : 29 using $\omega = 100$, consequently real time authentication becomes a realistic option.

The second advantage, again as demonstrated in the paper, is that the approach is more effective; a best overall accuracy of 97.86% was recorded with respect to the proposed mechanism as opposed to a best overall average accuracy of 95.51% using a univariate approach and 83.71% when an established feature vector technique was used (based on that presented in (Gunetti and Picardi, 2005)). For future work, the authors intend to investigate the use different time series representation methods, such as the Fourier transform, to determine the effectiveness of such different representation methods for real time KCA. Furthermore, the time complexity of DTW, in the context of keystroke multivariate time series similarity, remains an open topic for future work. We wish to optimise the operation of DTW with respect to such keystroke multivariate time series.

REFERENCES

Ahmed, A. A. and Traore, I. (2014). Biometric recognition based on free-text keystroke dynamics. *Cybernetics, IEEE Transactions on*, 44(4):458–472.

Alshehri, A., Coenen, F., and Bollegala, D. (2016a). Keyboard usage authentication using time series analysis. In *International Conference on Big Data Analytics and Knowledge Discovery*, pages 239–252. Springer.

Alshehri, A., Coenen, F., and Bollegala, D. (2016b). Towards keystroke continuous authentication using time series analytics. In *Proc. AI 2016, Research and Development in Intelligent Systems XXXIII, Springer, pp275-287.*, pages 325–338. Springer.

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *KDD workshop*, pages 359–370. Seattle, WA.

Bours, P. (2012). Continuous keystroke dynamics: A different perspective towards biometric evaluation. *Information Security Technical Report*, 17(1):36–43.

Cao, D. and Liu, J. (2016). Research on dynamic time warping multivariate time series similarity matching based on shape feature and inclination angle. *Journal of Cloud Computing*, 5(1):11.

Clark, R. C. and Mayer, R. E. (2016). *E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. John Wiley & Sons.

Dowland, P. S. and Furnell, S. M. (2004). A long-term trial of keystroke profiling using digraph, trigraph and keyword latencies. In *Security and Protection in Information Processing Systems*, pages 275–289. Springer.

Gaines, R. S., Lisowski, W., Press, S. J., and Shapiro, N. (1980). Authentication by keystroke timing: Some preliminary results. Technical report, DTIC Document.

Giancardo, L., Arroyo-Gallego, T., Butterworth, I., Mendoza, C., Montero, P., Matarazzo, M., and Sanchez-Ferro, A. (2016). Computer keyboard interaction as an indicator of early parkinson's disease. *arXiv preprint arXiv:1604.08620*.

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

Gunetti, D. and Ruffo, G. (1999). Intrusion detection through behavioral data. In *Advances in Intelligent Data Analysis*, pages 383–394. Springer.

Hu, B., Chen, Y., Zakaria, J., Ulanova, L., and Keogh, E. (2013). Classification of multi-dimensional streaming time series by weighting each classifier's track record. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 281–290. IEEE.

Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. *IEEETrans. Acoustics, Speech, and Signal Processing*, pages 52–72.

Joyce, R. and Gupta, G. (1990). Identity authentication based on keystroke latencies. *Communications of the ACM*, 33(2):168–176.

Killourhy, K. S. and Maxion, R. A. (2009). Comparing anomaly-detection algorithms for keystroke dynamics. In *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*, pages 125–134. IEEE.

Lines, J., Davis, L. M., Hills, J., and Bagnall, A. (2012). A shapelet transform for time series classification. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–297. ACM.

Polemi, D. (1997). Biometric techniques: review and evaluation of biometric techniques for identification and authentication, including an appraisal of the areas where they are most applicable. *Reported prepared for the European Commision DG XIIIC*, 4.

Raja, M. and Sigg, S. (2016). Applicability of rf-based methods for emotion recognition: A survey. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 1–6. IEEE.

Richardson, A., Kaminka, G. A., and Kraus, S. (2014). Reef: Resolving length bias in frequent sequence mining using sampling. *International Journal On Advances in Intelligent Systems*, 7(1):2.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization fro spoken word recognition. *IEEE Trans. Acoustics, Speech, and Signal Processing*, pages 43–49.

Shepherd, S. (1995). Continuous authentication by analysis of keyboard typing characteristics. In *Security and Detection, 1995., European Convention on*, pages 111–114. IET.

Syed, Z., Banerjee, S., and Cukic, B. (2014). Normalizing variations in feature vector structure in keystroke dynamics authentication systems. *Software Quality Journal*, pages 1–21.

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., and Keogh, E. (2003). Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 216–225. ACM.

Vural, E., Huang, J., Hou, D., and Schuckers, S. (2014). Shared research dataset to support development of keystroke authentication. In *Biometrics (IJCB), 2014 IEEE International Joint Conference on*, pages 1–8. IEEE.

Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., and Keogh, E. (2013). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309.