Behavioural Biometric Continuous User Authentication Using Multivariate Keystroke Streams in The Spectral Domain

Abdullah Alshehri[1][0000−0003−0008−9394], Frans Coenen[2][0000−0003−1026−6649], and Danushka Bollegala[2][0000−0003−4476−7003]

[1] Department of Computer Science, Albaha University, Saudi Arabia
aashehri@bu.edu.sa
[2] Department of Computer Science, University of Liverpool, United Kingdom
{coenen,danushka.bollegala}@liverpool.ac.uk

Abstract. Continuous authentication is significant with respect to many online applications where it is desirable to monitor a user's identity throughout an entire session; not just at the beginning of the session. One example application domain, where this is a requirement, is in relation to Massive Open Online Courses (MOOCs) when users wish to obtain some kind of certification as evidence that they have successfully competed a course. Such continuous authentication can best be realised using some forms of biometric checking; traditional user credential checking methods, for example username and password checking, only provide for "entry" authentication. In this paper, we introduce a novel method for the continuous authentication of computer users founded on keystroke dynamics (keyboard behaviour patterns); a form of behavioural biometric. The proposed method conceptualises keyboard dynamics in terms of a Multivariate-Keystroke Time Series which in turn can be transformed into the spectral domain. The time series can then be monitored dynamically for typing patterns that are indicative of a claimed user. Two transforms are considered, the Discrete Fourier Transform and the Discrete Wavelet Transform. The proposed method is fully described and evaluated, in the context of impersonation detection, using real keystroke datasets. The reported results indicate that the proposed time series mechanism produced an excellent performance, outperforming the comparator approaches by a significant margin.

Keywords: Biometrics · Continuous Authentication · Keystroke Dynamics · Keystroke Time Series

1 Introduction

Recent decades have seen a considerable increase in the popularity of digital learning. Digital learning, also referred to as online education and eLearning, refers to internet facilitated education, as opposed to traditional face-to-face classroom-style education. A current example is the prevalence of MOOCs (Massive Open Online Courses) where students learn at their own pace and withdraw

openly and freely [7]. The increasing popularity of digital learning, whatever form this might take, has resulted in an increasing number of people who wish to attain some kind of certification as evidence of successful completion of (say) an online programme. One mechanism for doing this is in the form online assessments and exams which students take remotely. Consequently, user authentication has become an issue [14, 25, 28]; certification providers need systems in place to confirm that the person taking an online assessment/exam is who they say they are.

Today, the vast majority of digital learning systems depend on traditional (*log-in*) credentials, such as passwords and usernames, for authentication. However, this means that the identity of students is only authenticated at the start of an eLearning assessment. The utilisation of this form of authentication is obviously inadequate with respect to what is known as "insider attacks". The form of insider attack most relevant to eLearning is impersonation, where an imposter poses as the real user when performing some kind of remote assessment. Therefore, a major issue with respect to digital learning systems is how to continuously confirm that a student taking an assessment is who they say they are. This means, not only that students need to be authenticated at the start of each assessment, but throughout the course of the assessment; continuous authentication of student identity is thus required.

Continuous authentication can best be realised using some forms of Biometric checking system [36], because such systems operate using features that are inherent to the user [33]. Moreover, Biometrics can produce strong authentication solutions comparing to other forms of authentication [4]. Biometrics, in general, can be categorised as follows:

1. **Physiological Biometrics:** Physiological biometrics are the organic characteristics of an individual. Well known examples include: (i) iris recognition [40], (ii) face recognition [32] and (iii) fingerprint recognition [26].
2. **Behavioural Biometrics:** Behavioural biometrics are concerned with the manner in which individuals perform certain tasks. Examples include: (i) keystroke dynamics (typing patterns) [30], (ii) mouse movement usage [1], (iii) voice recognition [20], (iv) handwriting recognition [38] and (v) gait (walking style) recognition [27].

Typically, the use of physiological biometrics requires specialised equipment to operate, such as iris, face or fingerprint recognition devices. However, in the context of the digital learning domain, it seems unreasonable to expect online students to purchase such equipment for authentication reasons. Furthermore, physiological biometrics are impractical for continuous authentication in that students need to re-conduct the biometric authentication periodically. In contrast, behavioural biometrics, seem well suited to continuous user authentication in the eLearning context, because they do not require dedicated devices which in turn make their deployment relatively straightforward. The most obvious behavioural biometric to be used in the context of digital learning is keystroke dynamics (typing patterns).

Keystroke dynamics is a promising behavioural biometric recognition mechanism that can provide the desired continuous authentication [6]. It offers the advantage that no special equipment is required such as in the case of continuous iris or fingerprint recognition. The intuition behind the use of keystroke dynamics is that individuals use keyboards in different manners regardless of what they are typing [16]. Thus such "typing rhythms", captured using keystroke dynamics, can be effectively used to authenticate keyboard users. In this context, typing behaviour can be expressed in the form of patterns made up of the keystroke timing attribute-values associated with: (i) flight times ($\mathcal{F}^t$) and (ii) key-hold times ($\mathcal{KH}^t$) [16]. The first is the time from the first key press to the last key release of n-grams; the second is the duration of holding down a key. An n-grams in this context is a sequence of n keyboard characters.

Keystroke dynamics have been studied, as a biometric technology, in the context of Keystroke Static Authentication (KSA) and in the context of Keystroke Continuous Authentication (KCA). The first, as the name implies, is directed at user authentication with respect to static (fixed) texts such as passwords, usernames, and pin numbers. Whilst KCA is directed at authentication in the context of free (arbitrary) text.

The most common existing mechanism for learning typing patterns, regardless of whether KSA or KCA is being considered, is founded on the feature vector representation where individual feature vectors describe individual typing templates [5, 22, 30, 41]. In this context, the feature vectors typically comprise statistical values representing keystroke timing data specific to certain n-grams. For instance, the mean and standard deviation of the flight time for certain di-grams. Authentication is then conducted by determining the similarity between stored feature vectors representing reference templates (profiles) which are known to belong to a specific user, and a previously unseen current profile that is claimed to belong to the same specific user. If the similarity falls below some predefined threshold, the user is declared to be who they say they are; otherwise the user will be "flagged up" as a potential imposter.

The feature vector representation has met with some success, particularly in the context of KSA. However, in the context of KCA, the construction of typing templates, and the consequent learning of typing patterns, has been found to be more challenging. This is because, by definition, the text to be considered is free and unstructured. This, in turn, means that typing templates need to be much more generic than in the case of KSA (where we know what is going to be typed), and consequently more sophisticated. One approach is to generate feature vector templates by identifying statistical details concerning the most frequently occurring n-grams ([11, 17, 29]). However, a criticism that can be directed at this mechanism is that the generated typing templates (profiles) might not feature the same frequently occurring n-grams as the sample to be authenticated, which in turn can lead to poor authentication rates. A suggested solution is to increase the number of training n-grams considered; however, this means that users need to be asked to provide a lot of samples. An obvious question is how many n-grams do we require to ensure that a typing template is sufficiently

robust? Whatever the answer, the number of n-grams, and hence the number of required samples, is significant. Furthermore, feature vectors comprised of very large numbers of features raises efficiency concerns, particularly when the intention is to conduct continuous authentication, as required in the case of the online assessments and examinations frequently used in digital learning. Thus, a robust, accurate and more efficient continuous authentication mechanism, founded on keystroke dynamics, is desirable.

In this paper, a novel method for KCA is proposed using time series analysis to recognise typing patterns from free text in a manner suited to the continuous authentication required with respect to digital learning. The proposed method operates using, simultaneously, the $\mathcal{F}^t$ and $\mathcal{KH}^t$ keystroke timing features associated with all keystrokes, not just specific n-grams. More specifically, the proposed approach operates by considering typing behaviour in terms of Multivariate-Keystroke Time Series (M-KTS) subsequences of length ω. The idea is that these subsequences are extracted from a continuous keyboard dynamic stream, $\mathcal{K}_{ts} = \{p_1, p_2, \dots\}$, where each point p_i is a keystroke event represented in terms of $\mathcal{F}^t$ and $\mathcal{KH}^t$ values. The collected M-KTS subsequence are then transformed form the temporal domain to the spectral domain using either: (i) the Discrete Fourier Transformation (DFT) or (ii) the Discrete Wavelet Transform (DWT). In this manner, a spectral M-KTS can be obtained. KCA is then performed by comparing the most recent spectral M-KTS subsequence with the previous extracted spectral M-KTS subsequence (in a given typing session). On start-up the subject's identity will be initially confirmed in a "traditional" manner with reference to stored typing templates. The time series comparison is conducted using Dynamic Time Warping (DTW); a well-known similarity comparison method for time series.

The work presented in this paper is founded on previous work presented in [2] and [3]. In [2] KCA was accomplished using M-KTS but in the temporal domain, whilst in [3] KCA was realised using the spectral domain but only in the context of for Univariate Keystroke Time Series (U-KTS). The central intuition of the work presented in this paper was firstly that better KCA results could be obtained using M-KTS than were obtained using U-KTS (regardless of whether the time series are considered in the temporal or spectral domain). Secondly that usage of the spectral domain would be better suited to KCA because with respect to other time series applications, such as time series indexing [13] and time series pattern extraction [35], it had been demonstrated that usage of the spectral domain could significantly improve analysis in terms of both speed and accuracy.

The remainder of this paper is structured as follows. Section 2 reviews the pertinent previous work concerning KCA. Section 3 presents some preliminaries for the multivariate keystroke time series representation. The framework for the proposed KCA method, using spectral M-KTS, is then presented in Section 4. Next the evaluation of the proposed approach is reported in Section 5. Finally, the work is summarised and concluded in Section 6.

2 Previous Work

As noted in the introduction to this paper, there has been significant previous work directed at KCA, although directed at the use of statistical measurements to define feature vectors with respect to sets of n-grams. One of the earliest reported studies can be found in [29] where typing templates were constructed using feature vectors comprised of $\mathcal{F}^t$ mean values for all di-grams that featured in a training set. KCA was performed by repeatedly generating "test" feature vectors for a given user, one every minute, and comparing these with stored templates. If a statistically similar match was found this was considered to be a correct authentication. A criticism of this approach is the size of the feature vectors to be constructed because of the large number of di-grams that were needed, and thus the search complexity was expensive. To minimise the search complexity, the authors proposed a clustering mechanism, so only the most relevant cluster had to be searched in detail. However, this then meant that re-clustering was required every time a new user was added. Furthermore, a reported accuracy of only 23% was reported.

In [11], $\mathcal{F}^t$ was also used for the construction of feature vectors. Each feature vector was generated by considering the first 500 di-grams and tri-grams in the input typing sample, and the most frequently occurring 2000 keywords in the English language. Valid $\mathcal{F}^t$ values were required to be within the range 10ms to 750ms. The mean and Standard Deviation (SD) of each di-gram, tri-gram and keyword were extracted and n-grams with SD values in the top and bottom 10% pruned so as to remove n-grams that had very large or very small SDs. During KCA, potential imposter samples were compared with a stored template and an "alert criterion" adjusted accordingly. A deviation (threshold) value was used to identify imposters. For evaluation purposes, a simulated environment was used. The metric used to measure the performance of the system was the False Match Rate (FMR). Experiments were conducted using di-grams, tri-grams and keywords; independently and in combination. Best results were obtained using di-grams. The reason that tri-grams and keywords did not perform well was that the tri-grams did not appear as frequently as di-grams; many of the identified keywords did not appear at all in the test data.

The study presented in [17] utilised the average $\mathcal{F}^t$ values for shared di-grams and tri-grams between two given typing samples. The similarity between two typing samples, determined for KCA purposes, was performed as follows. The $\mathcal{F}^t$ average values of all shared n-grams in the two samples were extracted, and ordered, in ascending order, in two arrays. The similarity was then computed by finding the differences between the order numbering of each n-grams in each array and summing them to give a "degree of disorder" value. The smaller the degree of disorder the more similar the two typing samples. Thus, authenticating a new typing sample required comparison with all stored typing samples (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).

The work presented in [2] and [3] first considered the use of time series analysis in the context of KCA; the first using M-KTS but in the temporal domain, the second in the spectral domain but using U-KTS. In [2] and [3] it was conclusively demonstrated that time series-based approaches outperformed feature vector-based approaches. The work presented in this paper was motivated by the desire to improve on the work of [2] and [3], and by extension the feature vector based approach.

3 Preliminaries

The generic concept of time series is well defined in the literature (see for example [39]); however, this section presents the application of the concept to keystroke time series, more specifically M-KTS. The section commences with a formal description of what a keystroke time series is and then goes on to define the keystroke timing features used.

Definition 1. *A keystroke time series, $\mathcal{K}_{ts}$, is an ordering of keyboard events $\{p_1, p_2, \ldots, p_n\}$ 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 key-hold time ($\mathcal{KH}^t$). That is, each event $p_i \in \mathcal{K}_{ts}$ can be given as:

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

such that a keystroke time series can be formulated as an M-KTS of the form

$$\{\langle t_1, \mathcal{F}_1^t, \mathcal{KH}_1^t \rangle, \langle t_2, \mathcal{F}_2^t, \mathcal{KH}_2^t \rangle, \ldots\}$$

Because the identifying index t can be inferred from the ordering of points in the time series, the M-KTS can be simply conceptualised as a series of dimensional points such that:

$$\{\langle \mathcal{F}_1^t, \mathcal{KH}_1^t \rangle, \langle \mathcal{F}_2^t, \mathcal{KH}_2^t \rangle, \ldots\}$$

Given a keystroke time series $\mathcal{K}_{ts_i}$ (in the form of M-KTS) of length n, it can be divided in to $\frac{n}{\omega}$ subsequences where $\omega \in \mathbb{N}$ and $1 < \omega \leq n$ is the length of the derived subsequences.

Definition 3. *A keystroke time series subsequence (s), of length ω, is a subsequence of $\mathcal{K}_{ts}$ that starts at the point p_i within $\mathcal{K}_{ts}$ and ends at point $p_{i+\omega-1}$, thus:*

$$s = \{p_i, p_{i+1}, \ldots, p_{i+\omega-1}\}$$

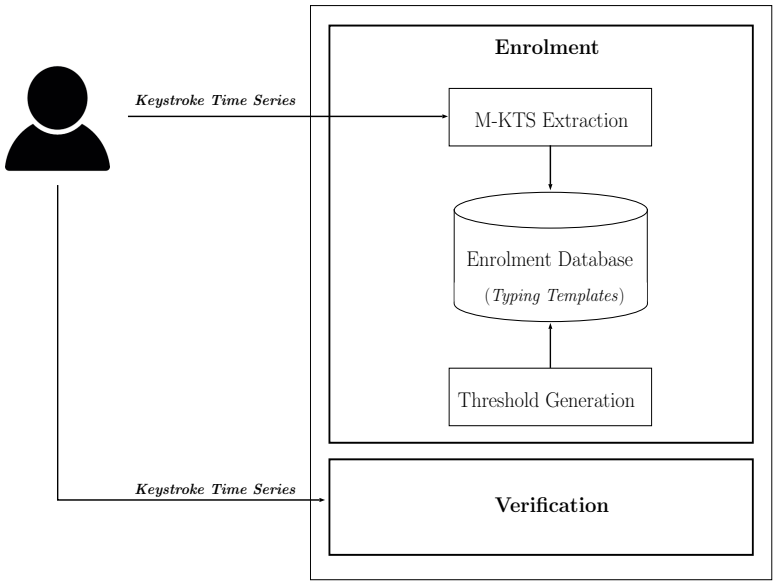


Fig. 1. The proposed KCA operational framework.

4 Framework of The Proposed KCA Method

The proposed KCA method operates, at a high-level, in a similar manner to other biometric pattern recognition mechanisms in that it features two central components, enrolment and verification, as shown in Figure 1. Enrolment is the process whereby an enrolment database, a database of typing templates for legitimate users, is built up. The enrolment stage also involves the generation of individual threshold values (σ values) for each subject. Verification is then the process whereby subjects are authenticated. The first precedes the second.

In more detail, the fundamental components of the proposed KCA method can be subdivided into the following parts:

1. M-KTS Extraction.
2. Noise Reduction.
3. Transformation.
4. Similarity Comparison.
5. Template Construction.
6. Authentication.

Each of these is considered in further detail in the following sub-sections.

4.1 M-KTS Extraction

A key aspect of the proposed KCA method, with respect to both enrolment and verification, is the usage of M-KTS subsequences. The idea is that M-KTS subsequences are periodically extracted from the input data stream using a sliding

window of length ω, where ω is user-defined. More formally, given a keystroke time series $\mathcal{K}_{ts_i} = \{p_1, p_2, \ldots, p_n\}$, where p_i represents a typing event in the form of a $\mathcal{F}^t$ and $\mathcal{KH}^t$ pair, an M-KTS subsequence, s, of length ω, is periodically extracted such that $s = \{p_i, \ldots, p_{i+\omega-1}\}$. In this manner, an ordered collection of M-KTS subsequences is produced, $\{s_1, s_2, \ldots, s_k\}$. The extracted M-KTS subsequence can then be used either as user typing templates or for authenticate purposes. It was anticipated that a small ω value would provide efficiency gains, desirable in the context of KCA; whilst a larger value would provide for accuracy gains. A trade-off between efficiency and accuracy was therefore anticipated.

4.2 Noise Reduction

An issue with keystroke time series represented using $\mathcal{F}^t$ values is that these capture significant pauses in keyboard activity and, on occasion, "away from keyboard" events. It was found that such pauses could adversely affect the extraction of typing patterns from keystroke time series. The problem is illustrated in Figure 2 where a time series, indicated using a black-dotted line, is shown featuring 300 keystrokes and $\mathcal{F}^t$ values where some of the values are significantly higher than the rest. From the figure, it can be seen that there is significant fluctuation in the amplitude of the curve, fluctuation which was found to impede the effectiveness of any time series analysis applied. To address this issue, it was decided to apply some data cleaning to the keystroke time series stream as it arrived so that data with abnormally high $\mathcal{F}^t$ values, in other words "noise" or "outlier" values, could be removed.

To this end, a threshold, φ, for acceptable values of $\mathcal{F}^t$ was defined. The idea was to use this threshold, not to remove points from time series subsequences, but to reduce the associated $\mathcal{F}^t$ value to the value of φ where $\mathcal{F}^t > \varphi$. Returning to Figure 2 the red time series indicates the same time series as the black-dotted time series but with a φ threshold of 2 seconds applied. In the context of the proposed KCA, the above was applied to each M-KTS subsequence, s, as it was collated. The pseudo code presented in Algorithm 1 describes the noise reduction process. The inputs are: an M-KTS subsequence, s, where $s \subseteq \mathcal{K}_{ts}$ and the points represent keystroke feature tuples; and a φ value. The output is a subsequence $\widehat{s}$ with $\mathcal{F}^t$ values greater than φ reduced to the value of φ.

The question that remains is what the value of φ should be. This is considered in further detail in Section 5 where the results from a series of experiments are reported on using a range of values for φ from 0.75 to 2.00 seconds increasing in steps of 0.25 seconds ($\{0.75, 1.00, 1.25, 1.50, 1.75, 2.00\}$).

It should also be noted that key-hold time, $\mathcal{KH}^t$, is normally no longer than 1 second. Inspection of the datasets used in this thesis indicated that the highest recorded value of $\mathcal{KH}^t$ was 950 millisecond. Consequently, it was felt that no threshold needed to be applied to $\mathcal{KH}^t$ values as in the case of $\mathcal{F}^t$ values.

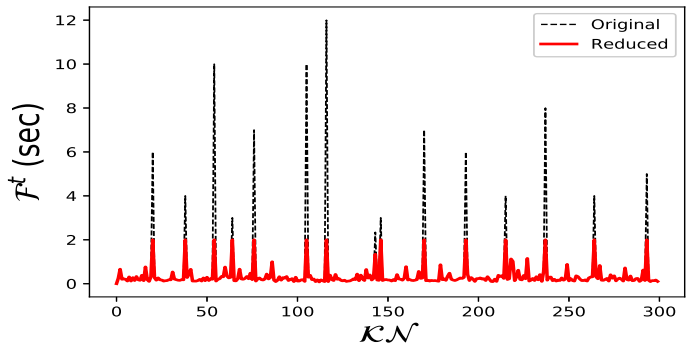


Fig. 2. The effect of applying a threshold φ to a keystroke time series stream $\mathcal{K}_{ts}$ so as to limit the $\mathcal{F}^t$ values, $\varphi = 2$ (sec).

Algorithm 1 Reducing Outlier Values of $\mathcal{F}^t$.

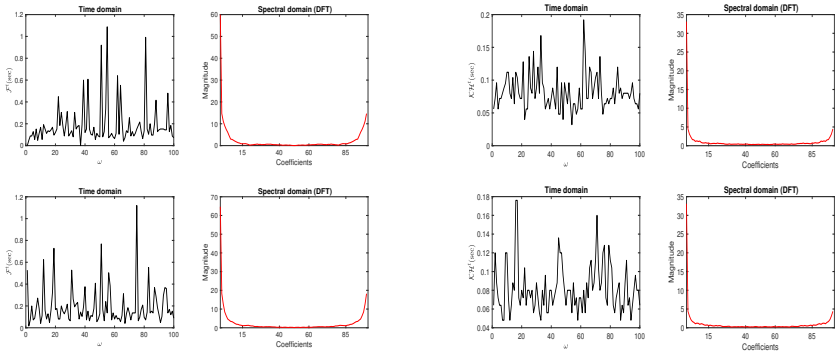
Input: $s \leftarrow$ subsequence of $\mathcal{K}_{ts}$, $\varphi \leftarrow \mathcal{F}^t$ limit.
Output: $\widehat{s} \leftarrow$ subsequence with reduced $\mathcal{F}^t$.

1: $s \leftarrow (p_1, p_2, \ldots, p_i, \ldots, p_l)$
2: $l \leftarrow$ length of s
3: **for** $i = 1$ to $i = l$ **do**
4: $\quad p_i \leftarrow \langle \mathcal{F}_i^t \rangle$ ▷ Return $\mathcal{F}^t$ value from ρ (a tuple point).
5: $\quad$ **if** $p_i > \varphi$: **then**
6: $\quad\quad p_i == \varphi$
7: $\quad\quad$ Update(s)
8: $\quad$ **end if**
9: **end for**
10: **Return** $\widehat{s}$

4.3 Transformation

The next component of the proposed KCA method was the transformation of the extracted M-KTS sequences from the temporal domain to the spectral domain. As noted in the introduction to this paper, two spectral transforms were considered: (i) Discrete Fourier Transformation (DFT) and (ii) Discrete Wavelet Transform (DWT). Both are considered in further details in the following two sub-sections (see also considered in [3]).

The Discrete Fourier Transform for Keystroke Streams. The fundamental idea of the DFT is to transform a given M-KTS subsequence from the temporal domain into the frequency domain. The resulting frequency-domain representation shows how much of a given signal lies within each given frequency band over a range of frequencies. The fundamental benefit is that the DFT serves to compact the data without loosing any salient information [21].



(a) (b)

Fig. 3. Examples of the application of DFT for subject **A**; (**a**) $\mathcal{F}^t$ keystroke subsequences, (**b**) $\mathcal{KH}^t$ keystroke subsequences.

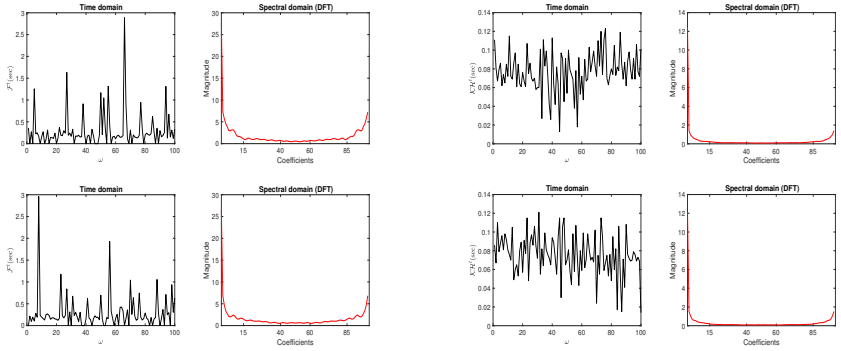
Compression is conducted by first representing the time series as a linear combination of sinusoidal coefficients, and then computing the similarity between the transformed coefficients for any pair of corresponding signals. Given an M-KTS subsequence, $s = \{p_1, p_2, \ldots, p_i, \ldots, p_\omega\}$, where p_i is some keystroke timing feature, and ω is the length of the subsequence, the DFT transform typically compresses the subsequence s into a linear set of sinusoidal functions X with amplitudes p, q and phase w, such that:

$$X = \sum_{i=1}^{\omega} (p_i Cos(2\pi w_i p_i) + q_i Sin(2\pi w_i p_i)) \tag{1}$$

Note that the time complexity to transform (each) s is $\mathcal{O}(\omega \texttt{ log } \omega)$ using the Radix 2 DFT algorithm [10, 21].

Using the DFT transform the obtained keystroke subsequence s is composed of a new magnitude (the amplitude of the discrete coefficients) and phase spectral shape, which can be compared with other transformed keystroke time series subsequences.

Figures 3 and 4 illustrate the intuition behind the DFT as applied to M-KTS, within the context of the proposed KCA method, for typing samples associated with two subjects; Figure 3 for subject **A** and Figure 4 for subject **B**. Typing samples were taken from the ACB evaluation dataset presented in Section 5. Each figure comprises two subfigures: (**a**) $\mathcal{F}^t$ subsequences, and (**b**) $\mathcal{KH}^t$ subsequences. In each subfigure, two free text typing samples are shown, on the left-hand side the raw time series, and on the right-hand side the DFT equivalent time series. Form the figures, it can be seen that the DFT signals describe distinctive patterns of typing behaviour for the same subject.



(a) (b)

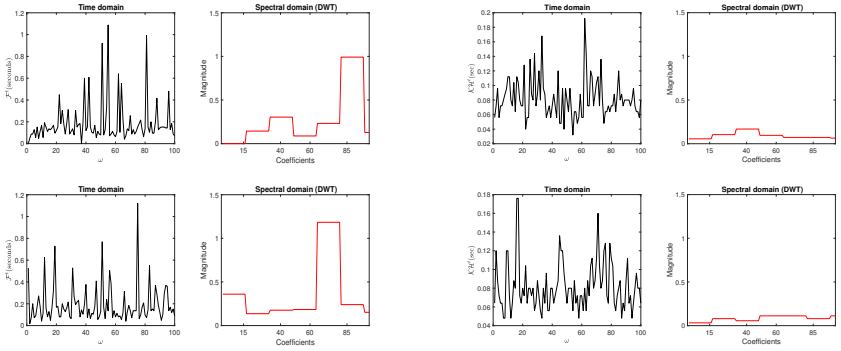
Fig. 4. Example of the application of DFT for subject **B**; (**a**) $\mathcal{F}^t$ keystroke subsequences, (**b**) $\mathcal{KH}^t$ keystroke subsequences.

The Discrete Wavelet Transform for Keystroke Streams. The Discrete Wavelet Transform (DWT) is an alternative form of time series representation that considers time series according to the frequencies that are present. DWT is sometimes claimed to provide a better transformation than DFT in that it retains more information [9]. DWT can be applied to time series according to different scales, orthogonal [18] and non-orthogonal [15]. For the work presented in this chapter the orthogonal scale was used, more specifically the well known Haar transform [18] as described in [9]. Fundamentally a Haar Wavelet is simply a sequence of functions which together form a wavelet comprised of a series of square shapes. The Haar transform is considered to be the simplest form of DWT; however, it has been shown to offer advantages with respect to time series analysis where the time series features sudden changes. The transformation is usually defined as shown in Equation 2 where, in the context of this thesis, x is a keystroke timing feature.

$$\phi(x) = \begin{cases} 1, & \text{if } 0 < t < \frac{1}{2} \\ -1, & \text{if } \frac{1}{2} < t < 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

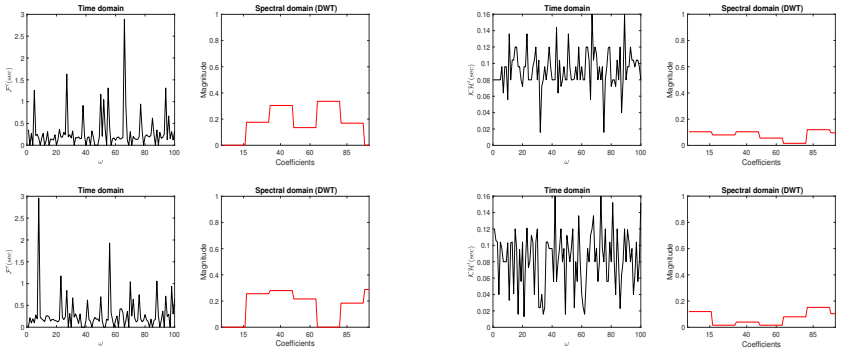
The time complexity for the Haar transform is $\mathcal{O}(n)$ for each $\mathcal{K}_{ts}$. Note that in the context of the Haar transform, the length of a given time series should be an integral power of 2 [23], thus 2, 4, 8, 16 and so on. For further detail concerning the DWT interested readers are to referred to [8] and [12].

The principle of DWT, as adopted with respect to the proposed KCA method, is illustrated in Figures 5 and 6 (in a manner similar to Figures 3 and 4). The figures show the DWT coefficients for keystroke subsequences obtained from two subjects, **A** and **B**; the same keystroke subsequences as given in Figures 3 and



(a) (b)

Fig. 5. Example of the application of DWT for subject **A**; (**a**) $\mathcal{F}^t$ keystroke subsequences, (**b**) $\mathcal{KH}^t$ keystroke subsequences.



(a) (b)

Fig. 6. Example of the application of DWT for subject **B**; (**a**) $\mathcal{F}^t$ keystroke subsequences, (**b**) $\mathcal{KH}^t$ keystroke subsequences.

4. The figures clearly show that DWT coefficients are distinctive in the context of keystroke data from the same subjects.

4.4 Similarity Comparison

Dynamic Time Warping (DTW), a well-established method for time series similarity checking, was adopted for the proposed KCA method. A great advantage of DTW is that it serves to warp the linearity of sequences (even of different lengths) so that any phase shifting can be taken into consideration. This is done

by calculating what is referred to as a *warping path*. The length, Θ, of this warping path (the minimum warping distance) is then treated as a similarity measure; if the length is zero the two time series under consideration are identical. Thus, it can be usefully adopted to find similarity in shape between two corresponding time series signals.

The method for determining Θ, using DTW, adopted with respect to the work presented in this paper, directed at M-KTS, is to calculate two warping paths. This could be achieved using two DTW matrices. However, it is more efficient to use a single matrix with two values stored in each cell. An alternative approach would have been to store 3-D distances at each cell. Although much less storage would be required to store such 3-D distances the calculation of 3-D distances would be equivalent to calculating two 2-D distances. The main advantage offered by the proposed two 2-D distances approach is simplicity.

Thus given two M-KTS sequences, such that $s_1 = \{p_1, p_2, \ldots, p_i, \ldots, p_x\}$ and $s_2 = \{q_1, q_2, \ldots, q_j, \ldots, q_y\}$, where x and y are the lengths of the two series respectively, and the values represented by each point $p_i \in s_1$ and each point $q_j \in s_2$ comprise a tuple of the form $\langle \mathcal{F}^t, \mathcal{KH}^t \rangle$, Θ is calculated as follows. First a DTW matrix $\boldsymbol{M}$ of size $(x-1) \times (y-1)$ is constructed. Each cell $m_{i,j} \in \boldsymbol{M}$ then holds two distance values, the difference between the $\mathcal{F}^t$ value for point $p_i \in s_1$ and that for point $q_j \in s_2$; and the difference between the $\mathcal{KH}^t$ value for point $p_i \in s_1$ and that for point $q_j \in s_2$.

The matrix $\boldsymbol{M}$ is used to find two minimum warping distance (Θ) associated with two minimum warping paths, $\mathbb{P}_{\mathcal{F}^t}$ and $\mathbb{P}_{\mathcal{KH}^t}$. Each warping path is determined as a sequence of cell locations, $\mathbb{P} = \{k_1, k_2, \ldots\}$, such that given $k_n = m_{i,j}$ the follow on location k_{n+1} is either $m_{i+1,j}$, $m_{i,j+1}$ or $m_{i+1,j+1}$. The value for a single Θ associated with a particular $\mathbb{P}$ is then the sum of the values held at the locations in $\mathbb{P}$:

$$\Theta = \sum_{n=1}^{|\mathbb{P}|} k_n \in \mathbb{P} \tag{3}$$

Consequently, two minimum warping distances are then determined, $\Theta_{\mathcal{F}^t}$ and $\Theta_{\mathcal{KH}^t}$. The final value for Θ is then the average of these two values:

$$\Theta = \frac{1}{2}(\Theta_{\mathcal{F}^t} + \Theta_{\mathcal{KH}^t}) \tag{4}$$

4.5 Template Construction

As previously indicated, the proposed KCA method operates using an enrolment database, a "bank" of subject (user) typing templates (profiles), one per subject. A user typing template $\mathcal{U}^T$ is therefore a set of m spectral M-KTS subsequences such that $\mathcal{U}^T = \{s_1, s_2, \ldots, s_m\}$. Note that the total length of the time series from which templates are generated must be substantially greater than the window size ω so that a significant number of M-KTS subsequences can be extracted. Figure 7 illustrates the process whereby a profile $\mathcal{U}^T$ is generated. In

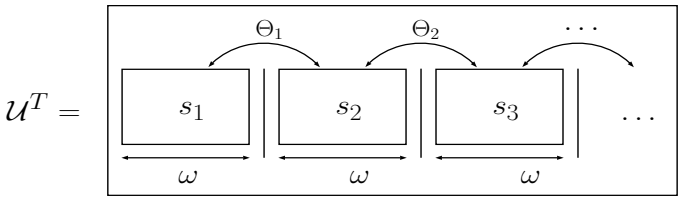


Fig. 7. A schematic illustrating the process of constructing a user typing profile $\mathcal{U}^T$ for a single subject.

the example, the windows are non-overlapping and abutting, this does not have to be the case, but this was the mechanism adopted with respect to the proposed KCA method evaluation presented later in this paper. The templates stored in the enrolment databases, as noted above, are also used to derive a bespoke similarity threshold, σ, for each user. This is calculated by comparing all spectral M-KTS subsequences within a template $\mathcal{U}^T$, using the DTW method described above, and obtaining an average warping distance $\bar{\Theta}$ which is then used as the value for σ:

$$\sigma = \bar{\Theta} = \frac{1}{|\mathcal{U}^T|} \sum_{i=2}^{|\mathcal{U}^T|} dtw(s_{i-1}, s_i) \tag{5}$$

It has been shown that averaging the warping distances associated with a set of time series can lead to an effective and more accurate classification of streaming data than if only one warping distance is considered [31].

4.6 Authentication

The actual KCA, in the context of the proposed time series-based method, is conducted by comparing the most recent spectral M-KTS subsequence with the immediately preceding spectral M-KTS subsequence (extracted during the typing session). At the beginning of the session, the subject's identity is first confirmed; in other words, it is confirmed that the subject is who (s)he says (s)he is. This initial process is called "start-up" authentication. In this context, the start-up authentication is done by comparing, using DTW, the first spectral M-KTS subsequence collected, s_1, with the relevant user template profiles in $\mathcal{U}^T$ (stored in the enrolment database) and obtaining an average similarity value (minimum warping distance). If the average similarity value is less than or equal to σ, the validation process proceeds accordingly. Each subsequent spectral M-KTS subsequence s_k (where $k > 1$) is then compared with the preceding, previously collected, subsequence s_{k-1}, again utilising DTW. In this manner, changes in typing behaviour can be detected.

The operation of the proposed KCA process is presented, more formally, in the form of pseudo code in Algorithm 2. The algorithm takes as input: (i) the M-KTS subsequence (window) size ω, (ii) the similarity threshold σ (derived

Algorithm 2 The proposed KCA algorithm.

Input: ω, σ, φ.
Output: Continuous authentication commentary.

1: $counter = 0$
2: $\mathcal{K}_{ts} = \emptyset$
3: **loop**
4: **if** terminated signal received **then**
5: break
6: **end if**
7: $p =$ keystroke features (e.g. $\mathcal{F}^t$ and $\mathcal{KH}^t$)
8: **if** $(\mathcal{F}^t \in p) > \varphi$ **then**
9: $p = \varphi$ ▷ Noise reduction.
10: **end if**
11: $\mathcal{K}_{ts} = \mathcal{K}_{ts} \cup \langle counter, k \rangle$
12: $counter++$
13: **if** $REM(counter/\omega) == 0$ **then**
14: $s_i =$ M-KTS subsequence $\{\mathcal{K}_{ts_{counter-\omega}} \dots \mathcal{K}_{ts_{counter}}\}$
15: **if** $counter = \omega$ **then** ▷ Start-up situation
16: $Transform(s)$ ▷ Transform s to (DFT)/(DWT)
17: Start-up: authenticate s_i w.r.t $\mathcal{U}^T$ and σ, and report
18: **else**
19: Authenticate s_i w.r.t. s_{i-1} and σ, and report
20: **end if**
21: **end if**
22: **end loop**

as described above in Sub-Section 4.5) and (iii) a φ threshold for limiting the $\mathcal{F}^t$ feature. The process operates continuously, following a loop, until the typing session is terminated (the user completes the assessment, times out or logs-out) (lines 4-6). Values for p are recorded as soon as the typing session starts (line 7). Note that in the case of flight time the value will be checked, and if necessary reduced according to φ (lines 8 to 10). The p value is then appended to the time series $\mathcal{K}_{ts}$. The *counter* is monitored and M-KTS subsequences are extracted whenever ω keystrokes have been obtained. Each extracted subsequence s is then transformed into DFT or DWT as required. The first transformed time series subsequence ($s_1 \in \mathcal{K}_{ts}$), the start-up time series, is compared with the stored profile for the subject in question; while each subsequent subsequence s_i is compared, using DTW, with the previous s_{i-1} subsequence.

5 Evaluation

This section presents a review of the evaluation conducted with respect to the proposed KCA using spectral M-KTS. The central objectives of the evaluation were:

1. **Typing Template Construction Efficiency:** To determine the efficiency of constructing the typing templates (enrolment database) using the proposed KCA approach.
2. **Authentication Performance**: To evaluate the effectiveness of the proposed KCA, in terms of impersonation detection, using different values for ω (the sampling window size) and φ (the noise reduction threshold value).

Note that the proposed method was evaluated using different values for ω and φ to determine the effect of these parameters on the KCA. As indicated in Subsection 4.3, the DWT transform can only support time series data whose length is defined as an integral power of 2, thus for the evaluation the range of ω values considered was $\{16, 32, 64, 128, 256, 512\}$, where the range of φ values considered was $\{0.750, 1.00, 1.25, 1.50, 2.00\}$ seconds.

The evaluation was also aimed, in the context of the above objectives, at providing a comparison with the approaches to KCA as proposed in the study presented in [2] and [3]. Recall that in [2] KCA was accomplished using M-KTS in the temporal domain and in [3] using the spectral domain but in the context of U-KTS. For ease of presentation, the following terminology is used in the remainder of this section:

1. M-KTS: KCA using the temporal domain applied to M-KTS as proposed in [2].
2. U-KTS+DFT: KCA using DFT applied to U-KTS as proposed in [3].
3. U-KTS+DWT: KCA using DWT applied to U-KTS as proposed in [3].
4. **M-KTS+DFT**: KCA using DFT applied to M-KTS as proposed in this paper.
5. **M-KTS+DWT**: KCA using DWT applied to M-KTS as proposed in this paper.

For the evaluation two datasets were used, as described in [2], namely the ACB and VHHS datasets. Each dataset consisted of typing samples collected from real subjects typing free (unstructured) text. Table 1 presents a summary of the characteristics of the two datasets; the table is based from [2]. The table lists the number of subjects, the environment setting where typing samples were collected, the language used to type samples, and the average and standard deviation of the keystroke time series with respect to each entire data set. Also, for the evaluation, the records associated with each subject in the datasets were divided into two so that one-half could be used for enrolment (typing template generation) and the other for authentication (typing stream simulation). Thus, two-fold cross-validation was conducted, hence results presented below are average results from two cross-validations. The metrics used for the evaluation were: (i) Authentication accuracy (Acc.), (ii) False Match Rate (FMR) and (iii) False Non-Match Rate (FNMR). FMR and FNMR are the standard metrics used to measure the performance of Biometric systems [37], although some researchers, in the literature, have used the terms FMR (False Acceptance Rate) and FRR (False Rejection Rate) instead.

The results obtained with respect to the two evaluation objectives are discussed below in further detail, Sub-sections 5.1 and 5.2 respectively.

Table 1. Summary of Evaluation Datasets [2].

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

Table 2. Typing template generation complexity (seconds) for spectral M-KTS applied to KCA.

ω	DFT		DWT	
	ACB	VHHS	ACB	VHHS
16	0.013	0.012	0.021	0.022
32	0.022	0.023	0.042	0.043
64	0.051	0.035	0.071	0.052
128	0.076	0.065	0.098	0.071
256	0.095	0.089	0.122	0.094
512	0.102	0.099	0.132	0.105

5.1 Typing Template Construction Efficiency

The first evaluation objective was to analyse the processing time required to generate the enrolment databases, including the associated individual σ threshold value calculation. Table 2 presents the average run-time complexity (seconds) results obtained for the construction of the typing template for each subject (the average time required to create the typing template for a single subject). From the table, it can be seen that the time complexity increases as ω increases. This was to be expected, as noted in Sub-section 4.1, because the time complexity to compute the DTW increases as the value for ω increases. Nonetheless, the results presented in Table 2 demonstrate that the constructing of typing templates was extremly efficiency; the worst run-time was less than one second.

Figure 8 gives the run-time (seconds) results obtained with respect to the proposed KCA using spectral M-KTS compared with the results obtained using the KCA variations given in [2] and [3]. Figure 8 (**a**) shows the reported run-time results for the ACB dataset, whilst Figure 8 (**b**) shows the run-time results for the VHHS dataset. From the figure, it can be seen that, regardless of which KCA variation was used, in all cases, the run-time increased as ω increased. As noted earlier, this was to be anticipated because the DTW computation time increases as the ω value increases. Overall the template construction efficiency results indicated that when using the proposed KCA approach (with spectral M-KTS) efficiency gains were made over the other approaches, except in the case of U-KTS+DFT which produced the best run-time. Nevertheless, with respect to the proposed KCA approach, it can be observed from the figure that M-KTS+DFT produced better run-time results than M-KTS+DWT; thus M-KTS+DFT was more efficient than M-KTS+DWT.

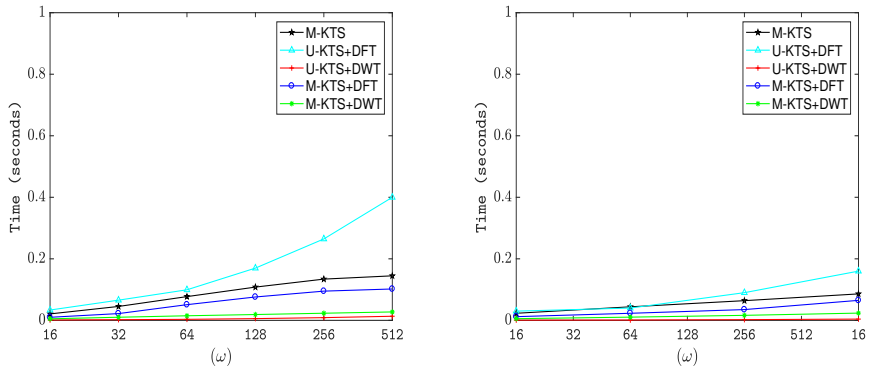


Fig. 8. Template construction run-time (seconds) comparison using variations of KCA: (**a**) ACB dataset, (**b**) VHHS dataset.

5.2 Authentication Performance

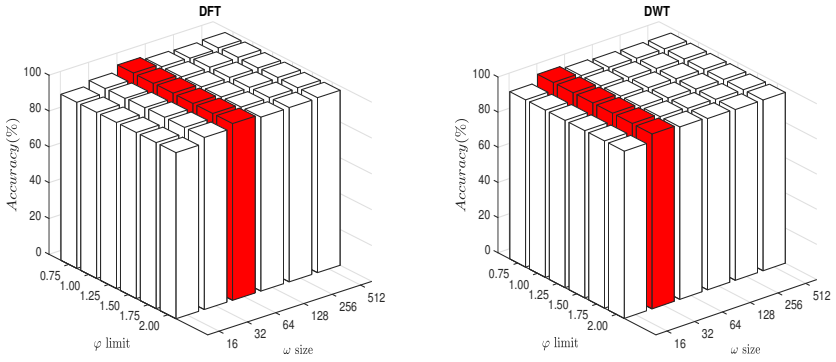
For each dataset, the continuous typing process was simulated by presenting the keystroke dynamics for each subject in the form of a data stream. In each case, the data stream was appended with a randomly selected second data stream from another user. The idea being to simulate one subject being impersonated by another half way through a typing session. For every comparison of a subsequence s_i with a subsequence s_{i-1}, it was recorded as to 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 6, 7 and 8).

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

$$FAR = \frac{FP}{FP + TN} \tag{7}$$

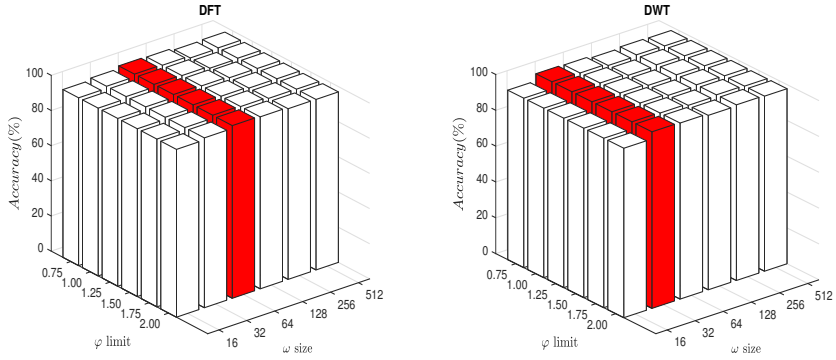
$$FRR = \frac{FN}{FN + TP} \tag{8}$$

The obtained accuracy results are given in the form of 3D bar charts in Figures 9 and 10 for the ACB and VHHS datasets respectively. In each figure, the vertical axis indicates accuracy, while the horizontal axises represent the window size (ω) and the limit value (φ). Each figure includes two such charts, with DFT on the left (**a**) and DWT on the right (**b**). From the figures, it can be observed that, in the context of M-KTS+DFT, best accuracy results were obtained when using $\omega = 64$ (with respect to both datasets); the red bars in the figure shows the best results with $\omega = 64$ across a range of φ values. However, in the context of M-KTS+DWT, best results were recorded at $\omega = 32$ across φ



(a) (b)

Fig. 9. The effect of ω and φ parameter settings on accuracy using the proposed KCA with **ACB** dataset.



(a) (b)

Fig. 10. The effect of ω and φ parameter settings on accuracy using the proposed KCA with **VHHS** dataset.

values. This means that good authentication accuracy can be gained using short time series subsequences. In other words, an accurate authentication can be obtained using only a small portion of the keystroke data stream; an important advantage for the form KCA desirable in the context of the online assessments frequently used with respect to digital learning. It can also be observed that when the value for ω increases beyond 64 the effect on accuracy is marginal. With respect to the φ parameter, the best recorded performance was obtained using $\varphi = 1.25$ seconds, although, it can be noted that the φ setting had less effect on authentication performance than the ω setting.

The accuracy (Acc.), FMR and FNMR results obtained, in the context of KCA coupled with spectral M-KTS, are summarised in tabular form in Table 3.

Note that in the context of DFT the reported results are shown when using $\omega = 64$ and $\varphi = 1.25$, whereas in the context of DWT the results are presented when $\omega = 32$ and $\varphi = 1.25$; the parameter values that produced the best results in each case. The table also gives the overall average values and the associated Standard Deviation (SD) in each case. The table clearly shows that DWT produced the best performance, with an average accuracy of 99.12% (and an associated SD of 0.77). For FMR and FNMR, the best obtained results were 0.010 and 0.816, again using DWT.

Table 3. Reported performance results (Acc, FMR and FNMR) using the proposed KCA approach.

Dataset	**DFT**			**DWT**		
	Acc.	FMR	FNMR	Acc.	FMR	FNMR
ACB	98.78	0.016	0.868	99.67	0.009	0.700
VHHS	98.30	0.018	0.941	98.58	0.011	0.932
Avg.	98.54	0.017	0.904	99.12	0.010	0.816
SD	0.34	0.002	0.051	0.77	0.001	0.165

For completeness, Tables 4 and 5 summarise the results obtained using the KCA variations in terms of accuracy, FMR and FNMR; Table 4 considers the ACB dataset, whilst Table 5 considers the VHHS dataset. In each case, the best performing ω and φ parameters wee used (also listed in the table). From the tables, it can be observed that the proposed spectral M-KTS with DWT (M-KTS+DWT) variation produced the best performance out of all the variations considered with respect to KCA in all metrics. The best accuracy was 99.67% with FMR and FNMR of 0.009 and 0.700 respectively for ACB dataset.

Table 4. Reported performance results (Acc, FMR and FNMR) for KCA variations applied to the **ACB** dataset.

Method	Acc.	FMR	FNMR	Best Parameters	
				ω	φ
M-KTS	98.39	0.045	1.093	125	1.50
U-KTS+DFT	97.43	0.130	1.500	64	1.25
U-KTS+DWT	99.22	0.029	1.070	64	1.25
M-KTS+DFT	98.78	0.036	1.091	64	1.25
M-KTS+DWT	**99.67**	**0.009**	**0.700**	32	1.25

From the foregoing, it can therefore be concluded that the proposed KCA method, using spectral M-KTS, provides a significant KCA improvement with respect to earlier time series-based KCA approaches.

Table 5. Reported performance results (Acc, FMR and FNMR) for KCA variations applied to the **VHHS** dataset.

Method	Acc.	FMR	FNMR	Best Parameters	
				ω	φ
M-KTS	97.32	0.057	1.095	125	1.50
U-KTS+DFT	97.42	0.045	1.085	64	1.25
U-KTS+DWT	97.09	0.059	1.098	64	1.25
M-KTS+DFT	98.30	0.018	0.941	64	1.25
M-KTS+DWT	**98.58**	**0.011**	**0.932**	32	1.25

6 Conclusion

In this paper, a novel method for Keystroke Continuous Authentication (KCA) has been presented. The idea was to use subsequences of keystroke streams in the form of Multivariate-Keystroke Time Series (M-KTS) of length ω. These subsequences incorporated both flight time $\mathcal{F}^t$ and key-hold time $\mathcal{KH}^t$ values. The idea was then to transform these subsequences from the temporal domain to the spectral domain. Two spectral transforms were experimented with: (i) Discrete Fourier Transform (DFT), and (ii) Discrete Wavelet Transform (DWT). The intuition was that such time series transformations would provide for efficiency gains and improved performance. Using the proposed KCA method, on start-up, the first spectral M-KTS subsequence extracted, s_1, for a given subject, is compared to a reference typing template. Then, the subsequence s_i ($i > 1$) will be compared to the immediate predecessor subsequence s_{i-1}, and so on. In this manner, continuous user authentication can take place. The comparison between transformed keystroke signals was conducted using Dynamic Time Warping (DTW) due to the advantages that DTW offered with respect to capturing time shifting (offsets) between corresponding subsequences.

The proposed KCA method was evaluated so as to establish its effectiveness and efficiency in the context of KCA. The evaluation also considered the effect of different parameter settings for the window size (ω) and the noise reduction limit (φ). The experimental results indicated that the proposed KCA, in the context of spectral M-KTS, coupled with the DWT spectral transform, outperformed KCA coupled with the DFT transform in terms of authentication performance; a best overall accuracy of 99.12% (with FMR = 0.010 and FNMR = 816) was recorded. In this context, the best result was obtained using $\omega = 32$ keystrokes, and $\varphi = 1.5$ seconds. However, the proposed KCA coupled with DFT was found to be the most efficient. Furthermore, it was observed that the proposed KCA method produced superior performance, in terms of authentication and efficiency, than the earlier KCA approaches presented in [2] and [3], and by extension the feature vector based approach from the literature.

For future work, the authors intend to investigate the performance of different time series transformations with respect to KCA. This is motivated by the observation that, in the proposed approach, a drawback of the Haar DWT transform is that the keystroke time series must have a length which is an inte-

gral power of two. An alternative is Piecewise Aggregate Approximation (PAA) [24] which operates using any time series length using an approximation of the DWT representation [23]. Consequently, the use of alternative time series transformations for the proposed KCA method is seen as a fruitfully topic for further research. Moreover, the time complexity of DTW, in the context of the proposed keystroke time series representation, remains an open research topic. From the literature a number of DTW mitigation techniques have been proposed (such as [19, 34]) which can provide for additional efficiency gains, these have yet to be investigated in the context of time series-based KCA.

References

1. Ahmed, A.A.E., Traore, I.: A new biometric technology based on mouse dynamics. Dependable and Secure Computing, IEEE Transactions on **4**(3), 165–179 (2007)
2. Alshehri, A., Coenen, F., Bollegala, D.: Accurate continuous and non-intrusive user authentication with multivariate keystroke streaming. In: Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 1: KDIR,. pp. 61–70. INSTICC, SciTePress (2017). https://doi.org/10.5220/0006497200610070
3. Alshehri, A., Coenen, F., Bollegala, D.: Spectral analysis of keystroke streams: Towards effective real-time continuous user authentication. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,. pp. 62–73. INSTICC, SciTePress (2018). https://doi.org/10.5220/0006606100620073
4. Asha, S., Chellappan, C.: Authentication of e-learners using multimodal biometric technology. In: Biometrics and Security Technologies, 2008. ISBAST 2008. International Symposium on. pp. 1–6. IEEE (2008)
5. Bergadano, F., Gunetti, D., Picardi, C.: User authentication through keystroke dynamics. ACM Transactions on Information and System Security (TISSEC) **5**(4), 367–397 (2002)
6. Bours, P.: Continuous keystroke dynamics: A different perspective towards biometric evaluation. Information Security Technical Report **17**(1), 36–43 (2012)
7. Breslow, L., Pritchard, D.E., DeBoer, J., Stump, G.S., Ho, A.D., Seaton, D.T.: Studying learning in the worldwide classroom: Research into edx's first mooc. Research & Practice in Assessment **8** (2013)
8. Burrus, C.S., Gopinath, R.A., Guo, H.: Introduction to wavelets and wavelet transforms: a primer. Prentice-Hall, Inc. (1997)
9. Chan, K.P., Fu, A.W.C.: Efficient time series matching by wavelets. In: Data Engineering, 1999. Proceedings., 15th International Conference on. pp. 126–133. IEEE (1999)
10. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex fourier series. Mathematics of computation **19**(90), 297–301 (1965)
11. Dowland, P.S., Furnell, S.M.: A long-term trial of keystroke profiling using digraph, trigraph and keyword latencies. In: Security and Protection in Information Processing Systems, pp. 275–289. Springer (2004)
12. Edwards, T.: Discrete wavelet transforms: Theory and implementation. Universidad de (1991)
13. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases, vol. 23. ACM (1994)

14. Furnell, S., Karweni, T.: Security issues in online distance learning. Vine **31**(2), 28–35 (2001)
15. Gabor, D.: Theory of communication. part 1: The analysis of information. Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering **93**(26), 429–441 (1946)
16. Gaines, R.S., Lisowski, W., Press, S.J., Shapiro, N.: Authentication by keystroke timing: Some preliminary results. Tech. rep., DTIC Document (1980)
17. Gunetti, D., Picardi, C.: Keystroke analysis of free text. ACM Transactions on Information and System Security (TISSEC) **8**(3), 312–347 (2005)
18. Haar, A.: Zur theorie der orthogonalen funktionensysteme. Mathematische Annalen **69**(3), 331–371 (1910)
19. Itakura, F.: Minimum prediction residual principle applied to speech recognition. IEEETrans. Acoustics, Speech, and Signal Processing pp. 52–72 (1975)
20. Jain, A., Hong, L., Pankanti, S.: Biometric identification. Communications of the ACM **43**(2), 90–98 (2000)
21. Janacek, G.J., Bagnall, A.J., Powell, M.: A likelihood ratio distance measure for the similarity between the fourier transform of time series. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 737–743. Springer (2005)
22. Janakiraman, R., Sim, T.: Keystroke dynamics in a general setting. In: Advances in Biometrics, pp. 584–593. Springer (2007)
23. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowledge and information Systems **3**(3), 263–286 (2001)
24. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Locally adaptive dimensionality reduction for indexing large time series databases. ACM Sigmod Record **30**(2), 151–162 (2001)
25. Maas, A., Heather, C., Do, C.T., Brandman, R., Koller, D., Ng, A.: Offering verified credentials in massive open online courses: Moocs and technology to advance learning and learning research (ubiquity symposium). Ubiquity **2014**(May), 2 (2014)
26. Maltoni, D., Maio, D., Jain, A., Prabhakar, S.: Handbook of fingerprint recognition. Springer Science & Business Media (2009)
27. Mantyjarvi, J., Lindholm, M., Vildjiounaite, E., Makela, S.M., Ailisto, H.: Identifying users of portable devices from gait pattern with accelerometers. In: Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on. vol. 2, pp. ii–973. IEEE (2005)
28. Moini, A., Madni, A.M.: Leveraging biometrics for user authentication in online learning: a systems perspective. IEEE Systems Journal **3**(4), 469–476 (2009)
29. Monrose, F., Rubin, A.: Authentication via keystroke dynamics. In: Proceedings of the 4th ACM conference on Computer and communications security. pp. 48–56. ACM (1997)
30. Monrose, F., Rubin, A.D.: Keystroke dynamics as a biometric for authentication. Future Generation computer systems **16**(4), 351–359 (2000)
31. Niennattrakul, V., Ratanamahatana, C.A.: Shape averaging under time warping. In: Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on. vol. 2, pp. 626–629. IEEE (2009)
32. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The feret evaluation methodology for face-recognition algorithms. IEEE Transactions on pattern analysis and machine intelligence **22**(10), 1090–1104 (2000)

33. Revett, K.: A bioinformatics based approach to user authentication via keystroke dynamics. International Journal of Control, Automation and Systems **7**(1), 7–15 (2009)
34. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization fro spoken word recognition. IEEE Trans. Acoustics, Speech, and Signal Processing pp. 43–49 (1978)
35. Staszewski, W.J., Worden, K., Tomlinson, G.R.: Time–frequency analysis in gearbox fault detection using the wigner–ville distribution and pattern recognition. Mechanical systems and signal processing **11**(5), 673–692 (1997)
36. Traore, I.: Continuous Authentication Using Biometrics: Data, Models, and Metrics: Data, Models, and Metrics. IGI Global (2011)
37. Unar, J., Seng, W.C., Abbasi, A.: A review of biometric technology along with trends and prospects. Pattern recognition **47**(8), 2673–2688 (2014)
38. Vielhauer, C., Steinmetz, R.: Handwriting: Feature correlation analysis for biometric hashes. EURASIP Journal on Applied Signal Processing **2004**, 542–558 (2004)
39. Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E.: Experimental comparison of representation methods and distance measures for time series data. Data Mining and Knowledge Discovery **26**(2), 275–309 (2013)
40. Wildes, R.P.: Iris recognition: an emerging biometric technology. Proceedings of the IEEE **85**(9), 1348–1363 (1997)
41. Wu, P.Y., Fang, C.C., Chang, J.M., Gilbert, S.B., Kung, S.: Cost-effective kernel ridge regression implementation for keystroke-based active authentication system. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. pp. 6028–6032. IEEE (2014)