Spectral Analysis of Keystroke Streams: Towards Effective Real-Time Continuous User Authentication

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: Continuous authentication using keystroke dynamics is significant for applications where continuous monitoring of a user identity is desirable, for example in the context of the online assessments and examinations frequently encountered in eLearning environments. In this paper, a novel approach to realtime keystroke continuous authentication is proposed that is founded on a sinusoidal signals based approach that takes into consideration the sequencing of keystrokes. Three alternative time series representations are considered and compared: Keystroke Time Series (KTS), Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT). The proposed process is fully described and analysed using three keystroke dynamics datasets. The evaluation also includes a comparison with the established Feature Vector Representation (FVR) approach. The reported evaluation demonstrates that the proposed method, coupled with the DWT representation, outperforms other approaches to keystroke continuous authentication with a best accuracy of 99.22%; a clear indicator that the proposed keystroke continuous authentication using time series analysis has significant potential.

1 INTRODUCTION

Keystroke dynamics are a form of behavioural biometrics which can be used to authenticate keyboard (keypad) users (Gaines et al., 1980; Alshehri et al., 2016b). Broadly, we can identify two forms of keystroke authentication: (i) static authentication and (ii) continuous authentication. The first is used in the context of one-time authentication, for example password or pin number access to a system; thus in the context of fixed texts. Some examples, from the literature, concerning this form of authentication can be found in (Bleha et al., 1990; Killourhy and Maxion, 2009; Syed, 2014). The second form of authentication is typically applied in the context of continuous free text where it is desirable to continuously monitor the identity of a user; examples regarding this form of authentication can be found in (Shepherd, 1995; Monrose and Rubin, 1997; Dowland and Furnell, 2004; Gunetti and Picardi, 2005; Ahmed and Traore, 2014). An application where continuous authentication is applicable is in the case of students completing online assessments as part of distance and online learning systems.

The focus of the work presented in this paper is continuous authentication. The reasons for this are as follows: (i) there is little reported work concerning continuous authentication using keystroke dynamics due to the challenges involved, and (ii) the increasing prevalence of internet facilitated distance learning (eLearning, Massive Open Online Courses and so on) where continuous authentication is desirable.

In this paper, we introduce a novel mechanism for keystroke continuous authentication, namely Keystroke Continuous Authentication based Spectral Analysis (KCASA) model. The proposed model is motivated by conceptualizing the process of keyboard usage as a continuous stream of keystroke events, thus as a time series which can be transformed into the spectral domain to extract typing patterns. More specifically, the idea is to convert a given keystroke stream from the temporal domain (raw data) to the sinusoidal (frequency) domain. The intuition is that such transformations for time series streams lead to faster, and more accurate, detection of patterns (Chan and Fu, 1999; Keogh et al., 2001). Therefore, keystroke streams can be effectively employed for real-time/continuous user authentication. In this study, two types of spectral transform are considered: (i) Discrete Fourier Transformation (DFT) and (ii) Discrete Wavelet Transform (DWT).

The remainder of this paper is structured as follows.

In Section 2, we provide a problem statement and discuss current issues with respect to keystroke continuous authentication. This is followed with Section 3 where definitions and preliminaries concerning the proposed model are given. Section 4 then discusses the proposed process of finding similarity between keystroke sinusoidal signals, while in Section 5 the proposed KCASA model is presented. The evaluation of the proposed approach is given in Section 6. Finally, the paper is concluded with a summary of the main findings and some recommendations for future work in Section 7.

2 PREVIOUS WORK

The fundamental approach of using keystroke dynamics for user authentication is founded on two keystroke timing features: (i) key hold time (KH^t), the elapsed time between a key press and a key release; and (ii) flight time (F^t), the time between n consecutive key presses (releases), also sometimes referred to as flight time latency or simply latency (Obaidat and Sadoun, 1997). Both can be indexed using either a temporal or a consecutive numeric reference. Whatever the case both flight time and hold time can be used to construct a distinctive typing profile associated with individual users (Gaines et al., 1980). These profiles are typically encapsulated using a feature vector representation of some form. In other words, typing profiles are frequently constructed using vectors of statistical values, such as the average and standard deviation of hold times, or the digraph flight time latency of selected frequently occurring digraphs. Authentication is then operated by comparing the similarity between stored feature vector represented typing (reference) profiles, which are known to belong to a specific user, and a previously unseen profile that is claimed to belong to a particular user. Although there has been only limited reported work directed at keystroke continuous keystroke authentication, what reported work there has been has used a feature vector representation; this has met with some success.

However, there are some limitations regarding the utilization of the feature vector representation in the context of keystroke continuous authentication. One of the main limitations is that the size of the feature vectors, a significant number of digraphs and/or trigraphs has to be considered which is infeasible in the context of real-time continuous authentication. In (Monrose and Rubin, 1997) the feature vectors were composed of the flight time means of all digraphs in the training dataset. The continuous authentication was then conducted by repeatedly generating "test" feature vectors for a given user, one every minute, and comparing with stored reference profiles. If a statistically similar match was found, then this was considered to be an indication of user authentication. Although the typing profile was composed of all digraph features, the overall reported accuracy was a surprising 23%. Similarly, in (Dowland and Furnell, 2004) the mean and Standard Deviation (SD) of the flight times for all digraphs and trigraphs in the training dataset were used. Thus, an average of 6,390 digraphs was needed to make up a sufficient typing profile.

Some researchers have attempted to use an abstraction of features to decrease the size of feature vectors. In (Gunetti and Picardi, 2005) the flight time, for frequent n-graphs, was used, although the approach was used in the context of user identification, as opposed to authentication. Thus, given a previously unseen sample, the shared n-graphs in the sample and the stored n-graphs were identified and collected in separate arrays. The elements in the arrays were then ordered according to flight time and the difference between the arrays computed by considering the orderings of the elements; a measure referred to as the *degree of disorder* was used (an idea motivated by Spearman's rank correlation coefficient (Zar, 1972)). Identifying a new sample required comparison with all stored sample profiles (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, construction typing profile using the average flight time of only shared n-graphs contained in the training data might not be representative of the n-graphs in the samples to be authenticated. This can, in turn, affect the authentication accuracy, especially in the context of real-time continuous authentication where typing patterns are extracted from free text; thus a substantial amount of n-graphs are expected to be typed in the current session. Furthermore, it can be observed from the study presented in (Gunetti and Picardi, 2005) that the authentication of one sample relies on all other samples in the training data. This can also lead to an efficiency issue where, in the context of continuous authentication, the current sample needs to be compared against the claimed user's reference profile.

In (Ahmed and Traore, 2014) an Artificial Neural Network classifier was used to build a prediction model to overcome the limitation of (Gunetti and Picardi, 2005) work. Key-down time was used together with average digraph and monograph flight times to predict missing digraphs based on the limited information in the training data; thus no need to involve a great number of keystroke features while constructing the typing profile. This mechanism worked reasonably well in the context of static authentication in a controlled setting (homogeneous); typing of the same text using the same keyboard layout in an allocated environment. Thus the work on continuous authentication remains an open area for further investigation. A general criticism of the feature vector approach is that the feature vector values are either typing pattern abstractions (for example average hold times) or only represent a subset of the data (for example only frequently occurring digraphs).

It is argued in this work that the feature vector representation may not be the most appropriate representation for keystroke continuous authentication. Therefore, it is conjectured that representing keystroke features as time series signals, and transforming these signals to the frequency domain, can lead to a better understating of typing patterns with respect to real-time continuous authentication using keystroke dynamics. To the best knowledge of the authors, no prior work in the literature has considered the concept of sinusoidal representation for keystroke dynamics in the context of continuous keyboard authentication. Note that in (Alshehri et al., 2016b) the authors first proposed the idea of keyboard continuous authentication using time series, but with respect to static text. In (Alshehri et al., 2016a) it was suggested that this could also be applied in the context of continuous text, although only hold time was considered. This paper presents a much more sophisticated realisation and analysis of the approach encompassing: (i) the idea of transforming the keystroke timing features into the sinusoidal (frequency) domain, (ii) using additional keystroke timing features to enhance the effectiveness of the authentication, (iii) usage of a transformed sinusoidal sliding windows to achieve the desired continuous authentication, (iv) a process for cleaning keystroke streaming data before authentication is conducted and (iv) a dynamic method for calculating similarity thresholds calibrated to individual users.

3 KEYSTROKE TIME SERIES REPRESENTATION

As already noted, the process of typing produces a Keystroke time series $K_{ts} = \{e_1, e_2, \ldots, e_n\}$ where e_n is an independent data event, and $n \in \mathbb{N}$ is the length of the time series. Each data event e_i represents a tuple of the form of $\langle t_i, k_i \rangle$ where: (i) t_i is a temporal index of some form, and (ii) k_i denotes some associated attribute (feature) value. Thus, $K_{ts} = \{\langle t_1, k_1 \rangle, \langle t_2, k_2 \rangle, \ldots, \langle t_i, k_i \rangle\}$. Such a time series can be viewed as a 2D plot with t along the x-axis and attribute value k along the y-axis (Figure 1). With respect to the work presented in this paper, the value for t_i is set to be a sequential ID number (sequence of key presses), whilst k records either flight time (F^t) or hold time (KH^t). Note that in this paper only univariate time series representation is considered, that is, in the evaluation section, we have adopted F^t and KH^t features independently to determine the effectiveness of each on the proposed model. Figure 1 shows four pairs of K_{ts} sequences, each featuring $n = 300$ keystrokes using F^t feature. The figure shows four (random) subjects selected from the datasets used for evaluation purposes as reported on in Section 6. Inspection of the figure clearly indicates that individual subjects have distinct keystroke streams and thus that they can be used to generate distinct typing profiles. Note that we also represent typing streams using KH^t, in the same manner; however, because of space limitations these are not included in the figure.

The generated keystroke time series can be used directly as described in (Alshehri et al., 2016b). However, as already noted, the usage of such "raw" time series is expensive in terms of efficiency and storage capacity (Agrawal et al., 1993). Thus the idea presented in this paper is to use some forms of transformation of the time series; it is conjectured that this will yield accurate results more efficiently. As noted in the introduction to this paper, two transformations are considered: (i) Discrete Fourier Transform (DFT), and (ii) Discrete Wavelet Transform (DWT). Each is discussed in further detail in the following two subsections.

3.1 DFT for Keystroke Time Series

The Discrete Fourier Transform (DFT) has been widely adopted with respect to time series data of all kinds (see for example (Agrawal et al., 1993; Vlachos et al., 2004)). In this paper, DFT has been used to transform keystroke time series data from the temporal domain to the frequency domain. The idea is then to compact the keystroke data points without losing any salient information. The compression is conducted by representing the keystroke stream as a linear combination of sinusoidal coefficients. Then the similarity is computed between the transformed coefficients for any pairs of corresponding signals.

Let's assume that we have a keystroke time series, such that $K_{ts} = \{e_1, e_2, \ldots, e_n\}$, where $k_i \in e_n$ is either a F^t or a KH^t value, and n is the length of the

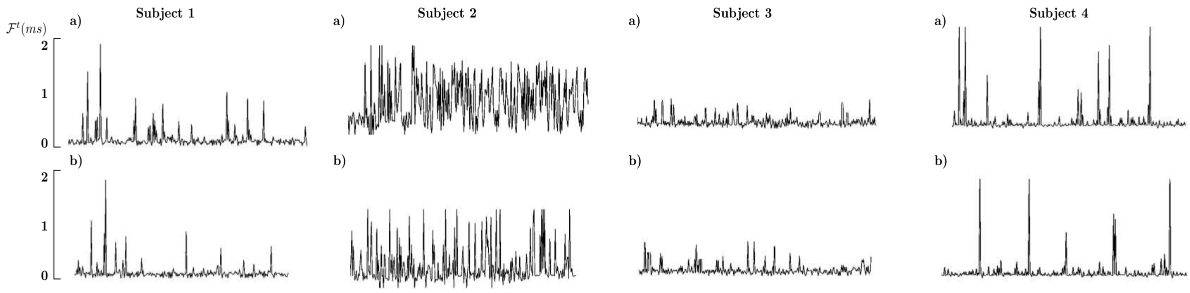


Figure 1: Examples time series ($n = 300$) for four subjects, two examples per subject, writing unspecified free text.

keystroke time series. The DFT transform then compresses K_{ts} into a linear set of sinusoidal functions with amplitudes p, q and phase w:

$$K_{ts} = \sum_{i=1}^{N} (p_i Cos(2\pi w_k F_i^t) + q_i Sin(2\pi w_i F_i^t)) \quad (1)$$

Note that the time complexity to transform (each) K_{ts} is $O(n \ \texttt{log} \ n)$ using the radix 2 DFT algorithm (Janacek et al., 2005; Cooley and Tukey, 1965).
Using the DFT transform, the obtained K_{ts} is composed of a new magnitude (the amplitude of the discrete coefficients) and phase spectral shape in which the similarity can be computed between pairs of transformed K_{ts} frequencies. Similarity measurement will be discussed in further detail in Section 4. For further detail concerning the DFT, interested readers are referred to (Harris, 1978).

3.2 DWT for Keystroke Time Series

The Discrete Wavelet Transform (DWT) is an alternative form of time series representations that considers the time span over which different frequencies are present in a time series. DWT is sometimes claimed to provide a better transformation than DFT in that it retains more information (Chan and Fu, 1999). DWT can be applied to time series according to different scales, orthogonal (Haar, 1910) and nonorthogonal (Gabor, 1946). In this paper, an orthogonal scale is used for the DWT, more specifically the well known Haar transform was adopted (Haar, 1910) as described in (Chan and Fu, 1999). 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 feature sudden changes. The transformation is usually described as per Equation 2 where, in the context of this paper, x is a keystroke timing feature.

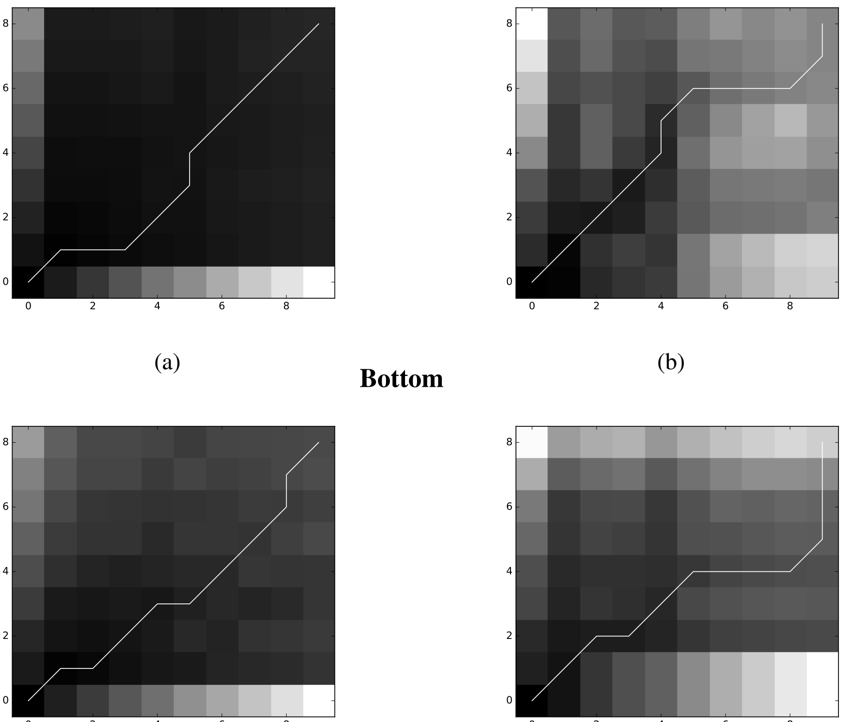
$$\phi(x) = \begin{cases} 1, & \text{if} \quad 0 < t < \frac{1}{2} \\ -1, & \text{if} \quad \frac{1}{2} < t < 1 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The time complexity for the Haar transform is $O(n)$ for each K_{ts}. For space limitations, we omit the full mathematical explanation of the Haar transform; however interested readers may refer to (Edwards, 1991) and (Burrus et al., 1997) for further detail.

4 SIMILARITY MEASUREMENT

To compare transformed keystroke streams, it is necessary to use some kind of similarity measure. Typically, given two keystroke streams (time series), S_1 and S_2 of the same length, the simplest way to compare them is to find the Euclidean absolute distances between all pairs of corresponding points in S_1 and S_2 and compute the average distance. If the average distance is 0, S_1 and S_2 are identical. However, this simple approach does not take into account offsets (phase shifts). For the process of the KCASA model, discussed in the following section, Dynamic Time Warping (DTW) was therefore adopted. The reason is that DTW takes into consideration phase shifting between pairs of signals more accurately than Euclidean distance measurement (Ye and Keogh, 2009).
DTW operates as follows. For two given (transformed) keystroke time series $S_1 = \{a_1, a_2, \ldots, a_i, \ldots, a_x\}$ and $S_2 = \{b_1, b_2, \ldots, b_j, \ldots, b_y\}$, where x and y are the length of the two series respectively, and (a_i and b_j) are DFT or DWT coefficients, the elements of each series are constructed in a matrix $\mathbf{M}$ of size $x \times y$. The value for each element $m_{ij} \in \mathbf{M}$ is then computed by calculating the distance from each element $a_i \in S_1$ to each element $b_j \in S_2$:

$$m_{ij} = \sqrt{(a_i - b_j)^2} \quad (3)$$



Top

(a) (b)

Bottom

(c) (d)

Figure 2: WPs examples. **Top:** WPs obtained from comparing two keystroke sinusoidal signals from same subject typing different texts, (a) DFT and (b) DWT. **Bottom** WPs obtained from comparing two keystroke sinusoidal signals from two different subjects writing different texts, (c) DFT and (d) DWT.

A Warping Path ($WP = \{m_{i_1,j_1}, m_{i_2,j_2}, \dots\}$) is then a sequence of matrix elements (locations), m_{ij}, such that each location is immediately above, to the right of, or above and to the right of, the previous location (see Figure 2). For each location, the next location is chosen so as to minimise the accumulated warping path length. The "best" warping path is the one that serves to minimise the distance from $m_{1,1}$ to $m_{x,y}$. The idea is then to find the path with the shortest "*warping distance*" (wd) between the two series calculated as follows:

$$wd = \sum_{i=1}^{i=|WP|} m_i \in WP \tag{4}$$

The value of wd is thus an indicator of the similarity between two keystroke signals; if $wd = 0$ the two keystroke signals are identical.

To further illustrate the concept of DTW, Figure 2 presents four WPs, resulting from application of the DTW process. Figures 2(a) and 2(b) show WPs obtained when DTW was applied to keystroke sinusoidal signals for the same subject writing different unknown texts; Figure 2(a) using DFT and 2(b) using DWT. In contrast, Figures 2(c) and 2(d) show the WPs obtained when comparing keystroke sinusoidal signals associated with two different subjects, writing different texts; Figure 2(c) using DFT and 2(d) using DWT.

5 KEYSTROKE CONTINUOUS AUTHENTICATION BASED SPECTRAL ANALYSIS (KCASA) OPERATION

The proposed KCASA model operates using a windowing approach, continuously sampling keystroke stream subsequences $K_w \subset K_{ts}$. The window size w is predefined by the user. Thus $K_w = \{e_i, e_{i+1}, \dots, e_w\}$ where i is a "start" time stamp. The keystroke stream subsequences can be made up of either flight time (F^t) or hold time (KH^t) values and can be processed simply as a straight forward time series, the Keystroke

Time Series (KTS) representation. Alternatively, as proposed in this paper, the time series can be transformed, using the DFT or DWT representation as described above. In the evaluation presented later in this paper, the effectiveness of the DFT and DWT representations is compared with the operation of the straight-forward KTS representation.

5.1 User Profile Calculation

A user profile $\mathcal{U}_p$ is a set of m non-overlapping keystroke streams (windows), or simply keystroke sinusoidal windows, $\mathcal{U}_p = \{W_1, W_2, \ldots, W_m\}$, where each window W has a length of ω. Note that $|\mathcal{U}_p|$ needs to be substantially greater than the window length ω, so that a number of subsequences (windows) can be extracted. Note also that the generated windows are prepared for the next transformation using DFT and DWT. Note also that ω is user defined. For the experiments reported on later in this paper, a range of ω values was considered from 25 to 150 key presses increasing in steps of 25, that is $\omega = \{25, 50, 75, 100, 125, 150\}$. By doing so, we can examine the effect of ω on performance in terms of accuracy. It was anticipated that a small window size would provide efficiency gains; that is desirable in the context of real-time continuous authentication.

The set $\mathcal{U}_p$ is also used to generate a bespoke σ threshold value. This is calculated by comparing all subsequences in $\mathcal{U}_p$ using DTW, and obtaining an average warping distance $\bar{wd}$ which is used as the value for σ:

$$\sigma = \bar{wd} = \frac{1}{|\mathcal{U}_p|} \sum_{i=2}^{|\mathcal{U}_p|} DTW(W_{i-1}, W_i) \qquad (5)$$

It has been shown that averaging the warping distances of time series lead to fast and accurate classification of streaming data (Niennattrakul and Ratanamahatana, 2009).

5.2 Subsequence Preprocessing and Noise Reduction

Before the KCASA authentication process can commence, each newly collated keystroke time series must be cleaned. The issue here is that F^t values can be large, for example when the subject has paused typing or as a consequence of (say) an "away from keyboard" event. A limit is therefore placed on F^t values using a maximum flight time threshold value φ. Given a F^t value in excess of φ, the value will be reduced to φ. For the evaluation presented later in this paper, a range of values for φ were considered, ranging from 0.750 to 2.00 seconds increasing in steps of 0.25 seconds, that is:

$$\varphi = \{0.75, 1.00, 1.25, 1.50, 1.75, 2.00\}$$

With respect to key hold time KH^t, the time whereby a key is held down is normally no longer than 1 second. Inspection of the datasets used in the study presented in this paper indicated that the highest recorded value of KH^t was 0.950 seconds. Consequently, it was felt that no maximum hold time threshold was required in this case.

5.3 The KCASA Algorithm

The pseudo code for KCASA process is presented in Algorithm 1. As already noted, the principle idea is, as typing proceeds, to collect non-overlapping keystroke sinusoidal windows, each of length ω, and compare these to previously obtained keystroke sinusoidal signals. On start up, it is first necessary to confirm that the user is who they say they are by comparing the first collected sinusoidal windows with the user profile $\mathcal{U}_p$ as described in Sub-section 5.1. As the session proceeds, continuous authentication is undertaken by comparing the most recent sinusoidal windows W_i with the previously collected sinusoidal windows W_{i-1}. Algorithm 1 takes the following inputs: (i) window size ω, (ii) a similarity threshold σ (derived as described above in Sub-Section 5.1) and (iii) a φ threshold for F^t. The process operates continuously in a loop until the typing session is terminated (the user completes the assessment, times out or logs-out) (lines 4-6). Values for k 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 replaced, according to φ (lines 8 to 10). The k value is then appended to the keystroke stream $\mathcal{K}_{ts}$. The $counter$ is monitored, and sub-sequences are extracted whenever ω keystrokes have been obtained. For the first collected window ($W_1 \in \mathcal{K}_{ts}$) this is the startup time series; each subsequent sinusoidal window W_i is then compared, using DTW, with the previous W_{i-1} sinusoidal window.

6 EVALUATION

A series of experiments were conducted to evaluate the proposed KCASA model to determine how well it performed in terms of the detection of impersonators. Comparisons were also undertaken with respect to a Feature Vector Representation (FVR), the established approach from the literature to keystroke con-

Table 1: Summary of datasets.

Dataset	# Sub.	Env.	Lang.	Features	Ave. size	SD
ACB	30	Free	English	F^t,KH^t	4625	1207
GP	31	Free.	Italian	F^t	7157	1095
VHHS	39	Lab.	English	F^t,KH^t	4853	1021

Algorithm 1 KCASA algorithm

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

$counter = 0$
$\mathcal{K}_{ts} = \emptyset$
loop
  if terminated signal received **then**
    break
  end if
  $k =$ keystroke feature (e.g. F^t or KH^t)
  if Flight time **&** $k > \varphi$ **then**
    $k = \varphi$ ▷ Noise reduction.
  end if
  $\mathcal{K}_{ts} = \mathcal{K}_{ts} \cup \langle counter, k \rangle$
  $counter++$
  if $REM(counter/\omega) == 0$ **then**
    W_i = subsequence $\{\mathcal{K}_{ts_{counter-\omega}} \dots \mathcal{K}_{ts_{counter}}\}$
    if $counter = \omega$ **then** ▷ Start up situation
      $Transform(W)$ ▷ Transform W to (DFT)/(DWT)
      Start up: authenticate W_i w.r.t $\mathcal{U}_p$ and σ, and report
    else
      Authenticate W_i w.r.t. W_{i-1} and σ, and report
    end if
  end if
end loop

tinuous authentication. The metrics used for the evaluation were: (i) authentication accuracy (Acc.), (ii) the False Acceptance Rate (FAR) and (iii) the False Rejection Rate (FRR)[1]. In more detail, the objectives of the evaluation were:

1. **Authentication Performance using the KCASA Model**: To compare the effectiveness of the DFT and DWT representations in the context of the proposed KCASA approach, and the usage of the simple KTS representation (as prposed in (Alshehri et al., 2016b)), in terms of accuracy, FAR and FRR.
2. **Effect on Authentication Performance using Different Parameters**: To determine the effect of using different values for ω (the sampling window size) and φ (the maximum flight time threshold value).
3. **Efficiency**: to compare the run time efficiency of KCASA in the context of the three representations considered (DFT, DWT and KTS).
4. **Comparison with Feature Vector Approach**: To compare the operation of KCASA with the established feature vector based approach for keystroke continuous authentication.

Note that the evaluation was conducted using flight time and hold time so as to also analyse which feature yielded the better results.
The rest of this section is organised as follows. The datasets used for the evaluation are introduced in Sub-section 6.1. The results with respect to the first set of experiments are considered in Sub-section 6.2, while those with respect to the second set of experiments in Sub-section 6.3. Efficiency is considered in Sub-section 6.4; and the comparison with the feature vector based approach is presented in Sub-section 6.5.

6.1 Datasets

Three datasets were used with respect to the reported experiments (Gunetti and Picardi, 2005; Vural et al., 2014; Alshehri et al., 2016b) to conduct. For ease of presentation the three data sets are identified here using acronyms made up of the authors' surnames: GP (Gunetti and Picardi, 2005), VHHS (Vural et al., 2014) and ACB (Alshehri et al., 2016b).
GP dataset comprised 31 subjects typing free text in Italian (that used in (Gunetti and Picardi, 2005) had 40 subjects, but some records are not available in the public version). The VHHS dataset was collected in laboratory conditions. The subjects were asked to type both predefined text and free text (in English); however, only the free text part was used with respect to the experiments reported on in this paper. Note also that for the GP dataset only the F^t feature was available, whilst for the remaining two datasets both F^t and KH^t were collected. Therefore the performance of KCASA using KH^t could not be evaluated using the GP dataset.
The ACB comprises 30 subjects although the original dataset consisted of 17 subjects, but the number

[1]FAR and FRR are the traditional metrics used to measure the performance of Biometric systems (Polemi, 1997).

of subjects has increased to 30 in the public version. Each subject provided free text samples (in English) in a simulated *online* assessment environment; the aim being to mimic the mode of typing when using an eLearning platform. Thus, the subjects used whatever keyboard they had at hand.

Table 1 provides a summary of the three datasets used; the table also includes some statical measurements concerning the average length of the time seres in each data collection and the associated Standard Deviation (SD). For evaluation purpose, each record in each data set was divided into two where the first half was used to generate the typing profile $\mathcal{U}_p$, and the second half for the continuous authentication evaluation.

6.2 Authentication Performance using the KCASA Model

The results obtained with respect to the evaluation directed at comparing the DFT, DWT and KTS KCASA representations, using either F^t or KH^t, are given in Tables 2 to 5; Tables 2 and 4 show the accuracy (Acc.), FAR and FRR results obtained using F^t, while Tables 3 and 5 presents the results, using the same metrics, obtained using KH^t. For the reported experiments, $\omega = 75$ keystrokes and $\varphi = 1.25$ seconds were used as default settings. These parameters were used because experiments, reported on in the following sub-section, had indicated that these produced best results.

From Table 2, it can be observed that the DWT representation produced the best overall accuracy (average accuracy of 98.24% with an associated Standard Deviation-SD of 1.07) when using F^t. With respect to FAR, we can observe from Table 4 that DWT also produced best results, except in the case of the GP datasets where DFT was recorded as producing the best result. It can also be noted from Table 4 that the DWT representation gave the best FRR results with an average of 1.50 and an associated SD of 0.14.

With respect to KH^t (Tables 3 and 5), a best accuracy results of 95.66% was obtained using DFT (with an associated SD of 2.40). Inspection of Table 5 shows that the best average FAR result was 0.04 when using the DFT representation, and the best average FRR result was 1.56 using DWT. Recall that evaluation using KH^t could not be conducted using the GP dataset because KH^t was not recorded in this case.

To support a comparison summary, the results listed in Tables 2 to 5 are presented in summary form in Table 6. From this summary table, it can be observed that the simple KTS representation did not perform well compared to the DFT and DWT representations. Also, from the results presented in this table, an argument can be made in favor of the DWT representation, coupled with F^t, which gave the best overall performance in terms of Acc, FAR and FRR.

6.3 Effect on Authentication Performance using Different Parameters

The results presented in the previous sub-section assumed a window size ω of 75 and a maximum F^t threshold value φ of 1.25. Recall that the latter is only applicable in the context of F^t. To evaluate the effect of these parameters, experiments were conducted using a range of values for ω and φ; $\{25, 50, 75, 100, 125, 150\}$ key presses for ω, and $\{0.75, 1.00, 1.25, 1.50, 1.75, 2.00\}$ seconds for φ.

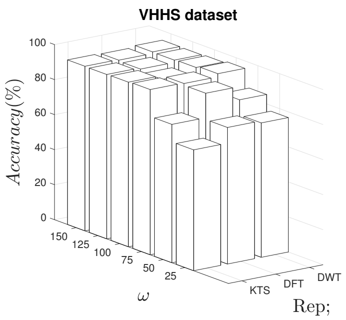


Figure 3: The effect of the ω parameter on accuracy using KH^t feature for VHHS dataset.

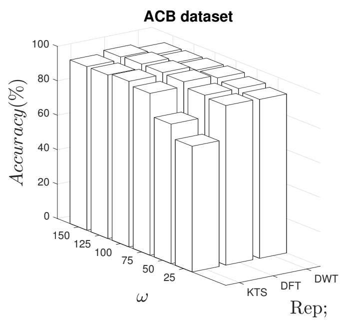


Figure 4: The effect of the ω parameter on accuracy using KH^t feature for ACB dataset.

The accuracy results using KH^t, as the keystroke dynamics, are shown in the form of 3D bar charts in Figures 3 and 4 for the VHHS and ACB datasets respectively. From the figure, it can be seen that $\omega = 75$ produced better accuracy results for the two datasets in terms of all three KCASA representations, with the exception of the KTS representation in the ACB dataset where $\omega = 100$ has produced better accuracy.

Table 2: Accuracy results obtained using the three different KCASA representations when using F^t (best results in bold font).

Dataset	Flight time F^t Accuracy		
	KTS	DFT	DWT
ACB	96.20	97.43	**99.22**
GP	95.47	96.94	**98.41**
VHHS	94.83	**97.43**	97.09
Average	95.50	97.27	**98.24**
SD	0.68	0.28	1.07

Table 4: FAR and FRR results obtained using the three different KCASA representations when using F^t (best results in bold font).

Dataset	Flight time F^t FAR			Flight time F^t FRR		
	KTS	DFT	DWT	KTS	DFT	DWT
ACB	0.050	0.030	**0.026**	1.96	1.50	**1.37**
GP	0.039	**0.034**	0.035	1.98	1.72	**1.48**
VHHS	0.030	0.022	**0.016**	1.97	1.85	**1.65**
Ave.	0.040	0.029	**0.026**	1.97	1.69	**1.50**
SD	0.010	0.006	0.010	0.01	0.17	0.14

Table 6: Summary of results presented in Tables 2 to 5.

Metric	F^t Feature			KH^t Feature		
	KTS	DFT	DWT	KTS	DFT	DWT
Acc	95.50	97.27	**98.24**	95.24	**95.66**	95.42
FAR	0.040	0.029	**0.026**	0.05	**0.04**	0.25
FRR	1.97	1.69	**1.50**	1.99	1.76	**1.56**

The accuracy results obtained using F^t as the keystroke dynamics are presented, again in the form of 3D bar charts, in Figure 5. From this Figure, it can be seen that ω and φ values of 75 and 1.25, respectively, tended to produce better results, although the selection of φ does not seem to have had as much impact as the selection of ω. Note also that accuracy "levels off" as ω is increased.

6.4 Efficiency

To compare the efficiency of the considered KCASA representations, experiments were conducted in terms of the time to generate the user profiles in each case. For the experiments, ω was set to a range of values, as described earlier, whilst φ was kept constant at 1.25 because earlier experiments, reported on above, had demonstrated that the value of φ was less significant. The efficiency performance using F^t is presented in Figure 6 with respect to each of the three datasets considered. From the Figure, it can be seen that as ω increased the run time also increased. This was to be expected because the computation time required for the DTW would increase as the size of the window ω increases. Interestingly, there are well-known solutions to mitigate against the complexity of DTW (see for example (Itakura, 1975; Sakoe and Chiba, 1978)); however, no such mitigation was applied with respect to the experiments reported on in this paper although this could clearly be done. We left this for further future investigation.

Table 3: Accuracy results obtained using the three different KCASA representations when using KH^t (best results in bold font).

Dataset	Key hold time KH^t Accuracy		
	KTS	DFT	DWT
ACB	96.15	**97.36**	95.09
VHHS	94.33	93.69	**95.75**
Average	95.24	**95.66**	95.42
SD	1.29	2.40	0.47

Table 5: FAR and FRR results obtained using the three different KCASA representations when using KH^t (best results in bold font).

Dataset	Key hold time KH^t FAR			Key hold time KH^t FRR		
	KTS	DFT	DWT	KTS	DFT	DWT
ACB	0.06	**0.04**	0.45	2.01	1.61	**1.38**
VHHS	0.03	**0.02**	0.04	1.97	1.91	**1.74**
Ave.	0.05	**0.04**	0.25	1.99	1.76	**1.56**
SD	0.02	0.01	0.29	0.02	0.22	0.25

Overall the results indicated that when using the proposed transformations efficiency gains were made with respect to the simple KTS representation, with DFT producing better runtime results than DWT. Furthermore, comparing the runtime performance obtained with the feature vector approach to keystroke authentication, it is interesting to note that in (Gunetti and Picardi, 2005) the time taken to construct a user profile was given as 140 second, a significant difference to when using the proposed KCASA method.

Note that in the context of KH^t, similar runtime results were produced to those presented in Figure 6, because both are using the same DTW similarity measure.

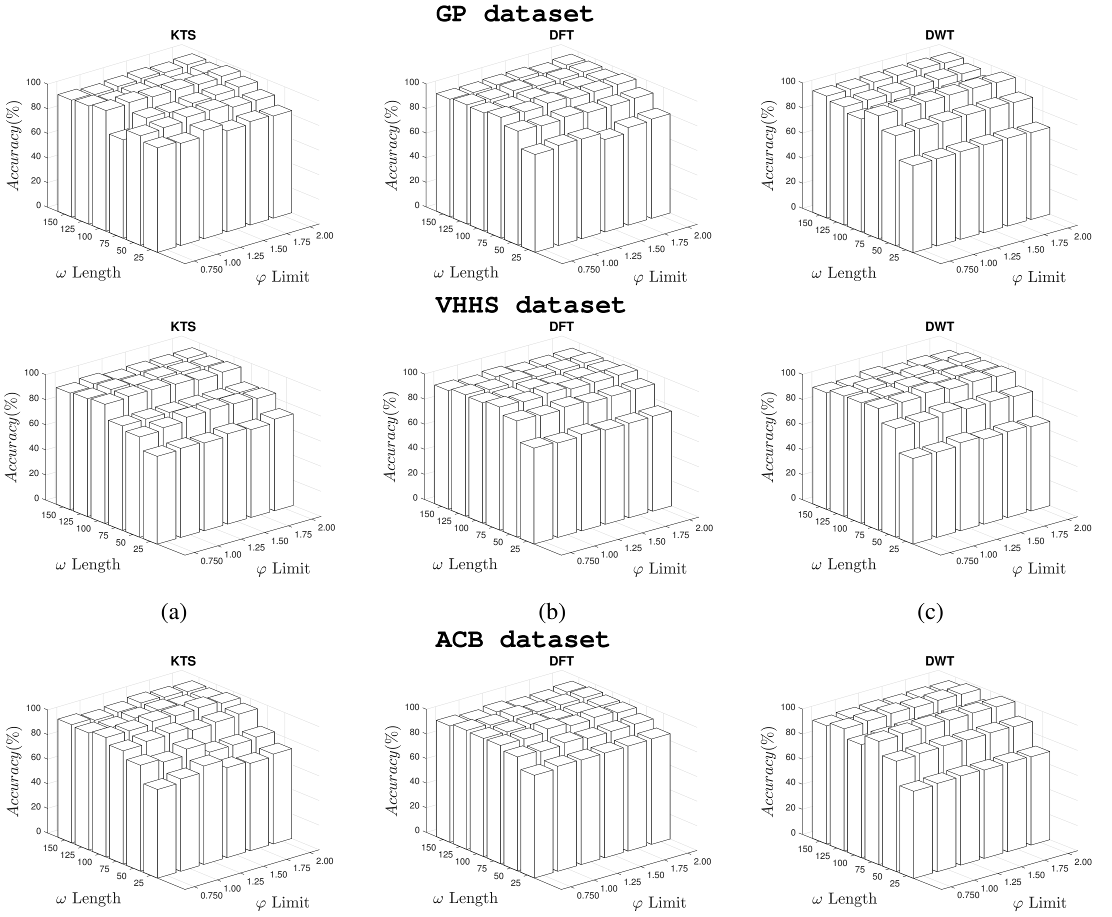


Figure 5: The accuracy results obtained for KTS, DFT and DWT using different values ω and φ.

6.5 Comparison with Feature Vector Approach

From the literature, previous work on keystroke continuous authentication has been conducted using Feature Vector Representation (FVR). It has already been noted that the proposed KCASA model has significant runtime advantages over the feature vector based approach (see above). However, it was felt appropriate to conduct further experiments comparing the operation of KCASA with the feature vector based approach in terms of authentication accuracy. Using both F^t and KH^t appropriate feature vectors were generated. Concequently, further comparison could be made with the mechanism proposed in (Gunetti and Picardi, 2005) (see Section 2). The reason for selecting the mechanism presented in (Gunetti and Picardi, 2005) was that the mechanism, to the best knowledge of the authors, had produced the best reported FAR and FRR results to date. However, it should be noted that the code for that mechanism is not available; thus we encoded the mechanism ourselves according to the description given in the original study. So as to conduct a fair comparison only F^t was considered, because the study in (Gunetti and Picardi, 2005) used F^t values. The average accuracy results obtained, when comparing the operation of FVR with the KTS, DFT and DWT representations, in terms of F^t, are given in Figure 7. The best accuracy result obtained for FVR was 90.15%, significantly worse than the accuracy results obtained using KCASA representations which yielded a best accuracy result of 98.24% (when using the DWT representation).

7 CONCLUSION

In this paper, a novel mechanism for realtime continuous keystroke authentication, called Keystroke Continuous Authentication using Spectral Analysis (KCASA) has been proposed, whereby authentication of user typing patterns is conducted by capturing keystroke dynamics in the form of spectral (fre-

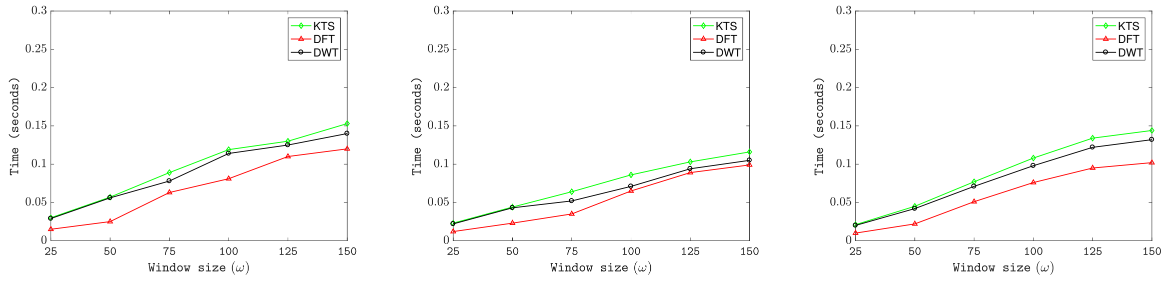


Figure 6: Runtime (seconds) comparison using flight time and the three KCASA representations with respect to each of the three datasets, (a) GP, (b) VHHS and (c) ACB.

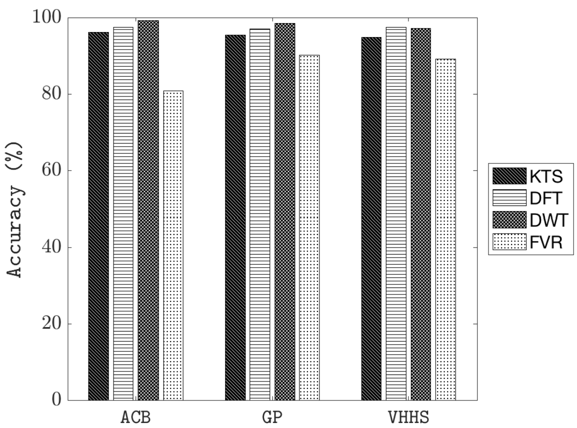


Figure 7: The obtained average accuracy using the three representations (KTS, DFT, DWT and FVR) with respect to the three datasets used. DWT shows a comparative performance with respect to KCASA model.

quency) streams. KCASA efficiently operates using either flight time F^t or hold time KH^t keystroke timing features. Two spectral transformations were considered to represent keystroke timing features: (i) Discrete Fourier Transform (DFT) and (ii) Discrete Wavelet Transform (DWT). Keystroke spectral streams similarity was conducted using Dynamic Time Warping (DTW), although alternative time series comparison techniques could equally well have been applied. The KCASA model operates by continuously extracting non-overlapped keystroke sinusoidal signals captured using a sliding window of size ω. The most appropriate size for ω was found to be 75 keystrokes for both timing features (flight time F^t and key hold time KH^t). In the case of flight time, an issue was discovered with excessive flight times; flight times were thus capped with a maximum value defined by a parameter φ, the most appropriate value for φ was found to be 1.25 seconds. The reported experimentation and evaluation indicated that the most accurate representation was DWT using the F^t keystroke feature, while the most efficient was found to be DFT. Experiments were also reported on indicating that the proposed KCASA model outperformed the feature vector based approach used by comparator systems such as that reported in (Gunetti and Picardi, 2005). For future work, the authors intend to investigate the usage of multivariate keystroke time series (incorporating F^t and KH^t timing features together) within the context of the proposed KCASA model. Furthermore, the time complexity of DTW, in the context of the proposed representations, remains an open topic for future work.

REFERENCES

Agrawal, R., Faloutsos, C., and Swami, A. (1993). Efficient similarity search in sequence databases. In *International Conference on Foundations of Data Organization and Algorithms*, pages 69–84. Springer.

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.

Bleha, S., Slivinsky, C., and Hussien, B. (1990). Computer-access security systems using keystroke dynamics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(12):1217–1222.

Burrus, C. S., Gopinath, R. A., and Guo, H. (1997). Introduction to wavelets and wavelet transforms: a primer.

Chan, K.-P. and Fu, A. W.-C. (1999). Efficient time series matching by wavelets. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 126–133. IEEE.

Cooley, J. W. and Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301.

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.

Edwards, T. (1991). Discrete wavelet transforms: Theory and implementation. *Universidad de*.

Gabor, D. (1946). 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.

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

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

Haar, A. (1910). Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371.

Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83.

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

Janacek, G. J., Bagnall, A. J., and Powell, M. (2005). 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*, pages 737–743. Springer.

Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. (2001). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems*, 3(3):263–286.

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.

Monrose, F. and Rubin, A. (1997). Authentication via keystroke dynamics. In *Proceedings of the 4th ACM conference on Computer and communications security*, pages 48–56. ACM.

Niennattrakul, V. and Ratanamahatana, C. A. (2009). Shape averaging under time warping. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on*, volume 2, pages 626–629. IEEE.

Obaidat, M. S. and Sadoun, B. (1997). Verification of computer users using keystroke dynamics. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 27(2):261–269.

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.

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. A. (2014). *Keystroke and Touch-dynamics Based Authentication for Desktop and Mobile Devices*. PhD thesis, West Virginia University.

Vlachos, M., Meek, C., Vagena, Z., and Gunopulos, D. (2004). Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 131–142. 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.

Ye, L. and Keogh, E. (2009). Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956. ACM.

Zar, J. H. (1972). Significance testing of the spearman rank correlation coefficient. *Journal of the American Statistical Association*, 67(339):578–580.