Hindawi
Mobile Information Systems
Volume 2018, Article ID 3127042, 15 pages
https://doi.org/10.1155/2018/3127042

Research Article

Online Signature Verification on MOBISIG Finger-Drawn Signature Corpus

Margit Antal, László Zsolt Szabó, and Tünde Tordai

Faculty of Technical and Human Sciences, Sapientia University, Cluj-Napoca, Romania

Correspondence should be addressed to Margit Antal; manyi@ms.sapientia.ro

Received 14 September 2017; Revised 8 November 2017; Accepted 3 December 2017; Published 14 February 2018

Academic Editor: Paolo Bellavista

We present MOBISIG, a pseudosignature dataset containing finger-drawn signatures from 83 users captured with a capacitive touchscreen-based mobile device. The database was captured in three sessions resulting in 45 genuine signatures and 20 skilled forgeries for each user. The database was evaluated by two state-of-the-art methods: a function-based system using local features and a feature-based system using global features. Two types of equal error rate computations are performed: one using a global threshold and the other using user-specific thresholds. The lowest equal error rate was 0.01% against random forgeries and 5.81% against skilled forgeries using user-specific thresholds that were computed a posteriori. However, these equal error rates were significantly raised to 1.68% (random forgeries case) and 14.31% (skilled forgeries case) using global thresholds. The same evaluation protocol was performed on the DooDB publicly available dataset. Besides verification performance evaluations conducted on the two finger-drawn datasets, we evaluated the quality of the samples and the users of the two datasets using basic quality measures. The results show that finger-drawn signatures can be used by biometric systems with reasonable accuracy.

1. Introduction

One of the oldest ways of proving your identity is giving your signature. Many official documents require signatures from agreeing parties. Signature recognition can be divided into off-line (static) and online (dynamic) methods. While off-line systems work with images, only the shape of the signature is available, but online systems use information related to the dynamics of the signature. Due to this additional information, online systems outperform off-line systems [1].

Biometric systems can produce two types of errors: false rejections of genuine signatures (false rejection rate (FRR)) and false acceptance of forged signatures (false acceptance rate (FAR)). The overall system error is usually reported in terms of EER (equal error rate), which is defined as the system error rate when FAR and FRR are equal.

In signature database evaluations, two types of forgeries are considered: skilled and random forgeries. Skilled forgery evaluation is based on using the forgery samples available in the database (forgery samples are provided by forgers who know both the shape and dynamics of the imitated signature). Random forgery (or zero effort) evaluation is based on using random genuine samples from the dataset (corresponding to the case when the forger does not know the signature to be forged, therefore is using his/her own signature). The state of the art in automatic signature verification is presented in a study by Impedovo and Pirlo [1].

Online signature recognition is not a new research area: several online signature corpora have already been collected using digitizer tablets. While PHILIPS [2], SVC2004 [3], and SUSIG [4] databases contain only online signatures, MCYT [5] is a bimodal database containing both fingerprints and online signatures. BIOMET [6] and BioSecurID [7] contain several types of biometric data including online signatures.

Due to the increasing number of touchscreen-based mobile devices and the familiarity of users with using signatures, we consider that signatures are plausible candidates for authentication on mobile devices. A number of researches have already been conducted in this topic, although the signature databases are not publicly available [8–10], except for DooDB database [11]. Compared to the DooDB database, which was collected on a device using a resistive touchscreen, our database was collected on a device with a capacitive touchscreen. Specifically, while DooDB contains

only the coordinates of the points of the signature and the corresponding time information, our MOBISIG database contains additional information such as pressure, finger area, and data saved from the accelerometer and gyroscope.

In this paper, we analyze whether signatures can be used for authentication in a mobile device context. Therefore, two state-of-the-art methods, a local- or function-based and a global- or feature-based system, were implemented and evaluated on the MOBISIG database. In order to compare our database to the other publicly available online signature database, we performed the same evaluations for the DooDB database using the same parameters and features. In addition, we present a few basic quality evaluations for both databases.

The main contribution of this paper is the presentation and analysis of MOBISIG signature database containing data from 83 users. The signatures are not the original signatures of the users, but users were assigned a family name and were required to create a signature for that name. Signature collection was performed on a Nexus 9 tablet under supervision; data providers were instructed on how to draw signatures using their finger. The database was collected during three sessions and contains 45 genuine and 20 forged signatures for each user. The database is publicly available at http://www.ms.sapientia.ro/~manyi/mobisig.html.

On the MOBISIG database, the best EER for skilled forgeries was obtained by our function-based DTW system: 5.81% for a posteriori user-specific thresholds and 20.82% for common thresholds. In case we added pressure information to the coordinates and their first- and second-order differences, only the a posteriori user-specific threshold result was improved. When using a 9-inch diameter device for data collection, users tend to put down the device on the table while drawing the signatures. Therefore, we did not use the data obtained from the accelerometer and gyroscope sensors in the computations.

Comparisons with the DooDB database indicate higher signature quality in the case of the MOBISIG database, and correspondingly better performances for the verification methods studied in our paper. Our study is limited by the sample size (83 users) and a slightly unbalanced age distribution (77% of the users aged under 25); in addition, our data providers were not experts in forging signatures.

The rest of the paper is organized as follows. A literature review on signature recognition on mobile devices is presented in Section 2, completed with a review of a few papers on signature quality evaluation. Section 3 presents our MOBISIG dataset, followed by a detailed description of the methods used for signature verification. Experiments and benchmark results are presented in Section 5, whereas Section 6 compares DooDB and MOBISIG datasets along verification system performance and quality measures. Section 7 concludes the paper.

2. Related Work

2.1. Signature Recognition on Mobile Devices. Little research has been carried out in the field of online signature recognition on mobile devices. We have only found six studies [8–13] reporting results obtained on signature databases captured in mobile context. The properties of the databases used in these studies are presented in Table 1.

In most of the studies concerning signature recognition, results are reported using signature databases captured on a pen tablet. However, touchscreens present some drawbacks compared to pen tablets, the most important being the quality of the captured signal. While pen tablets sample the signal uniformly with relatively high frequency, hand-held device sampling is usually event-driven with lower sampling frequency than pen tablets. Moreover, while both touchscreen devices and pen tablets are able to capture trajectory and pressure, the latter can track pen orientation. The only advantage of a touchscreen device is that it allows capturing the signature by fingertip.

One of the objectives of BioSecure Signature Evaluation Campaign (BSEC'2009) was to study the influence of acquisition conditions (digitizing tablet or PDA) on authentication systems' performance [14]. Results are reported using signatures from 382 writers, acquired on a digitizing tablet and on a PDA, respectively. The authors reported a significant quality degradation of signatures acquired in mobile conditions.

The semester thesis of Bissig [12] is the first study reporting results using a signature database captured on a resistive touchscreen with fingertip. Four types of signals were acquired: coordinates $x(t)$, $y(t)$, pressure $p(t)$, and finger area $a(t)$. Both local (function-based: DTW) and global systems (feature-based: one-class SVM and Mahalanobis distance) and the combination of these were evaluated. Unfortunately, neither the number of subjects nor the number of forgeries in the captured database is reported. However, this is the only study reporting the influence of pressure on the performance of a signature verification system.

Houmani et al. [8] report results on a new dataset collected from 64 subjects on a PDA. Unfortunately, neither the number of sessions nor the acquired signals are reported. However, they propose an entropy-based quality metric for selecting reference signatures in the enrollment phase.

Krish et al. [9] collected a new signature database using a Samsung Galaxy Note device from 25 users (20 genuine signatures from a user). Their verification algorithm combines two state-of-the-art algorithms (function-based DTW and feature-based Mahalanobis distance). Due to the missing forgery samples, only the results obtained by random forgery evaluations are reported.

Sae-Bae and Memon [10] collected an online signature dataset from 180 users using HTML5. Users were allowed to enter their signatures using their own iOS devices. The dataset is not publicly available and contains only genuine signatures; therefore, only the random forgery-type evaluation was feasible. They proposed a new histogram-based feature set and reported performance evaluations both on their and MCYT datasets.

The first publicly available database collected on a handheld device (HTC Touch HD mobile phone) is the DooDB. This database contains data from 100 users, and it also contains doodles besides pseudosignatures. Martinez-Diaz et al. [11] report the database analysis and benchmark results,

Table 1: Mobile signature databases.

Paper	Users	Device	Input Method	#GEN	#FOR	#SESS	Signals	Public
Bissig [12]	NA	HTC Desire 3.7', capacitive	Finger	20	NA	NA	$x(t)$, $y(t)$, $p(t)$, $fa(t)$	No
Houmani et al. [14]	432	PDA HP iPAQ hx2790	Pen	30	20	2	$x(t)$, $y(t)$	Yes
Houmani et al. [8]	64	PDA Qtek 2020 ARM	Pen	30	20	2	$x(t)$, $y(t)$	No
Krish et al. [9]	25	Samsung Galaxy Note	Pen	20	0	2	$x(t)$, $y(t)$, $p(t)$	No
Martinez-Diaz et al. [11]	100	HTC Touch HD mobile, resistive	Finger	30	20	2	$x(t)$, $y(t)$	Yes
Sae-Bae and Memon [10]	180	User-owned iOS devices	Finger	30	0	6	$x(t)$, $y(t)$	No
This paper	83	Nexus 9, capacitive	Finger	45	20	3	$x(t)$, $y(t)$, $p(t)$, $fa(t)$ $vx(t)$, $vy(t)$, $ax(t)$, $ay(t)$, $az(t)$ $gx(t)$, $gy(t)$, $gz(t)$	Yes

#GEN, number of genuine samples; #FOR, number of forgeries; #SESS, number of sessions; NA, unspecified.

using a function-based DTW verification system with several local features. Although the EERs obtained by the random forgery evaluations are low (around 3%), those obtained for skilled forgery evaluations are too high (around 27%). In a later study [13], the skilled forgery result was improved (20.9%) by using the Gaussian mixture method.

2.2. Signature Quality. Quality evaluation of biometric datasets is a difficult problem. A biometric dataset consists of biometric samples from a number of users, usually containing a fixed number of samples from each user collected in a fixed number of sessions. Moreover, signature datasets contain skilled forgery samples for each user.

There are two ways to achieve the quality evaluation of a biometric dataset: (i) evaluating each sample of the dataset and (ii) evaluating each user of the dataset. In each case, we obtain a set of scores, and then an average score can be computed from these scores. Both the samples and the users can be evaluated by using only the genuine signatures or by using both the genuine and forgery signatures. Müller and Henniger [15] proposed two quality metrics for signature dataset evaluation. One of the quality metrics evaluates the samples, while the other one evaluates the users of the dataset. Both metrics use the DTW distance between samples.

Houmani et al. [16] proposed a personal entropy measure for online signatures and showed the existence of a clear relationship between the proposed measure and the verification performance of the user revealed by the signature verification system. This measure allowed them to categorize the users of several signature datasets. In a later study [17], they adapted the measure to the skilled forgery samples of signature datasets. Similar to their previous study, they proved the effectiveness of the quality measure by evaluating several online signature databases by using state-of-the-art signature verification methods.

One of the objectives of the BSEC'2009 competition was the evaluation of online signature algorithms with respect to the quality of the signatures [14]. The personal entropy measure introduced by Houmani et al. was used to group the signatures into different categories. The results of the competition showed that the performance of classifiers varied significantly with respect to the good and bad quality signatures. Houmani and Garcia-Salicetti [18] extended the Biometric Menagerie to online signatures and categorized the users of MCYT database using the Personal Entropy quality measure.

Kahindo et al. [19] proposed a novel signature complexity measure to select reference signatures for online signature verification systems. Guest and Henniger [20] used commercial engines for the assessment of the quality of handwritten signatures. They concluded that predicting the utility of a signature sample using a multifeature vector was possible. More recently, another novel method was proposed for the quality evaluation of off-line signatures [21].

3. The MOBISIG Database

Due to security reasons (people are reluctant to give their own signatures), participants were asked to create a signature for a given family name. Family names were selected from the first 100 most frequent Hungarian family names. Participants were also asked to practice the created signatures by drawing and deleting several attempts. The first five attempts were deleted.

The database contains signatures from 83 subjects: 49 men and 34 women, with the following age distribution: 64 subjects under 25, 12 between 25 and 40, and 7 over 40.

3.1. Data Acquisition Protocol. Data collection was performed using a Nexus 9 tablet. The device has a capacitive touchscreen of $228.2 \times 153.7 \times 8$ mm ($8.98 \times 6.05 \times 0.31$ in.). Signatures were sampled at about 60 Hz (event-driven sampling) when the users pressed the screen. The resolution of the screen is 1536×2048 pixels (approx. 281 ppi pixel density). Each signature was stored as a sequence of discrete values $[x_t, y_t, t, p_t, fa_t, vx_t, vy_t, ax_t, ay_t, az_t, gx_t, gy_t, gz_t]$, where x_t, y_t are the coordinate values, t is the time stamp, p_t, fa_t are the pressure and finger area (these are normalized values [0, 1] and can be obtained through the standard Android API), vx_t, vy_t are the directional velocities, ax_t, ay_t, az_t are the directional acceleration of the device, and gx_t, gy_t, gz_t are the values obtained from the gyroscope

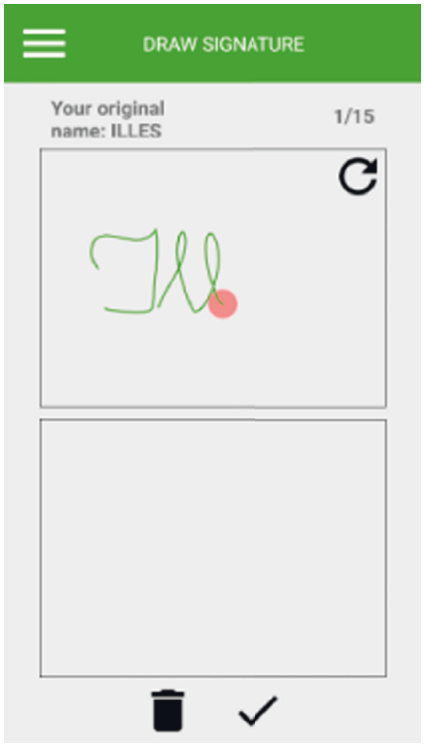


Figure 1: Data collection screen.

sensor. The accelerations and the values obtained from the gyroscope characterize the holding position of the device.

The screen of the device was divided into two sections (Figure 1): the upper section was the replay section, where users were shown the animated signature, and the lower section was designed to draw the signature. The animation functionality was available in both types of signature collections: genuine and forgery. The animation allows participants to recall the shape and the dynamics of genuine and forged signatures. The animation could be replayed any number of times. Before data collection, users were asked to become familiar with the device usage as well as their pseudosignatures. Additionally, signatures were saved after user's acceptance. Any signature could be deleted by the provider if he/she was not satisfied with the result.

The data collection process was divided into three sessions with one week between consecutive sessions. In the first session, each user had to provide 15 genuine pseudosignatures for the assigned name. In the second and third sessions, participants had to provide 15 genuine pseudosignatures and 10 forgeries for two assigned users (two times 5 forgeries). At the end of the data collection process, we had 45 genuine signatures and 20 forgeries for each participant. A few of these signatures are shown in Figure 2.

3.2. Signature Files. Each user has a dedicated folder which contains the 45 genuine signatures of the user and the 20 forgeries made by other users. The naming convention of the files is as follows: SIGN_[T] _USER[SID] _USER[WID] _[NR] , where T is FOR for forgeries and GEN for genuine signatures. SID identifies the user whose signatures are in the folder. WID is the identifier of the user, who gave the signature; SID and WID are equal in the case of genuine signatures, while they are different in the case of forgeries. The NR at the end of the filename is a value from 1 to 45 for genuine signatures and 1 to 20 for forgeries. The first 15 genuine signatures were collected in the first session, the second 15 signatures in the second session, and the third 15 signatures were collected in the third session. The first 10 forgeries were collected in the second session, while the second 10 forgeries in the third session. The naming conventions of the folders is USER[SID]. Each signature is represented as a sequence of points and is stored in a file. Each line of the file represents one point of the signature and consists of the following features: x-coordinate, y-coordinate, time stamp, pressure, finger area, x-velocity, y-velocity, x-acceleration, y-acceleration, z-acceleration, x-gyroscope, y-gyroscope, and z-gyroscope.

4. Methods

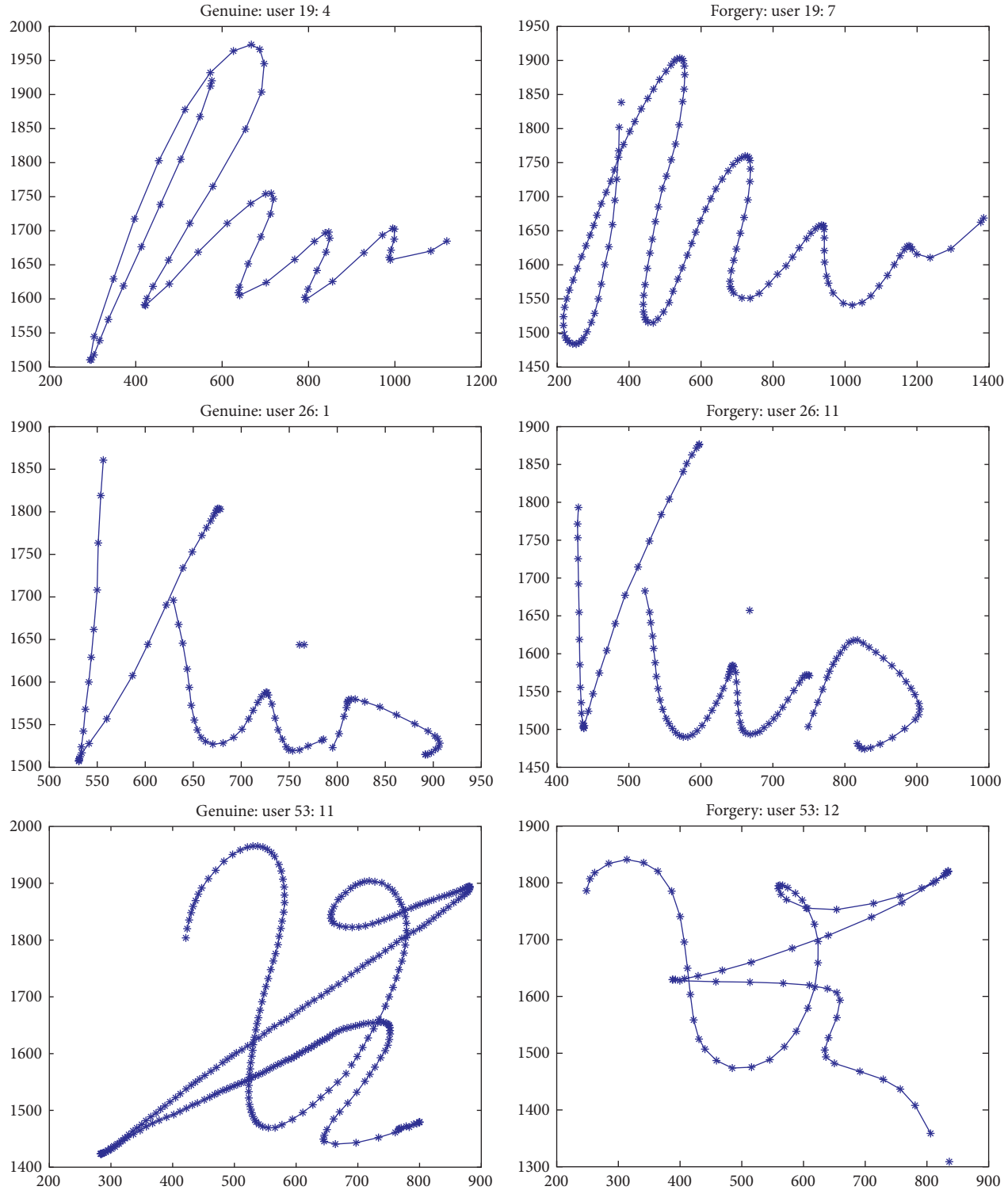
In order to assess the authentication performance based on pseudosignatures, both a function-based and a feature-based verification system were implemented. Features used by signature verification systems can be local and global ones. Local features correspond to sample points along the signature's trajectory (e.g.,point-wise pressure). Global features are computed from the signature as a whole (e.g.,duration). Function-based systems use local features and feature-based systems use global features.

4.1. Function-Based Verification. The first system is based on DTW (dynamic time warping), and it compares the captured time sequences $s[1 \ldots n]$ and $t[1 \ldots m]$ (Algorithm 1). Each signature $s[1 \ldots n]$ is represented as a time sequence with $s[j] = (f_1^j, f_2^j, \ldots, f_{N_f}^j)$, $j = 1, \ldots, n$, where N_f is the number of local features. In this work, the following local features were employed: the x, y coordinates, the x^1, y^1 first-order differences, the x^2, y^2 second-order differences, and the p, p^1 pressure and its first-order difference. Before computations, the features of time sequences were standardized $f_i' = (f_i - \mu_{f_i})/\sigma_{f_i}, i = 1, \ldots, N_f$, where μ_{f_i} and σ_{f_i} are the mean and standard deviation for the ith local feature computed over all sampling points of the signature). The Euclidean distance function was used to compute the distance between two elements of the time sequences (distance$(s[i], t[j])$). Finally, the obtained DTW distance was divided by the sum of the time sequence lengths $(n + m)$.

The verification process works as follows: in the enrollment stage, a set of N reference signatures are selected $\{e_1, e_2, \ldots, e_N\}$. In the verification stage, the DTW distances between the test signature and all the reference signatures are computed, and the final score results as the average of these distances. Finally, this distance-based score (Dscore)

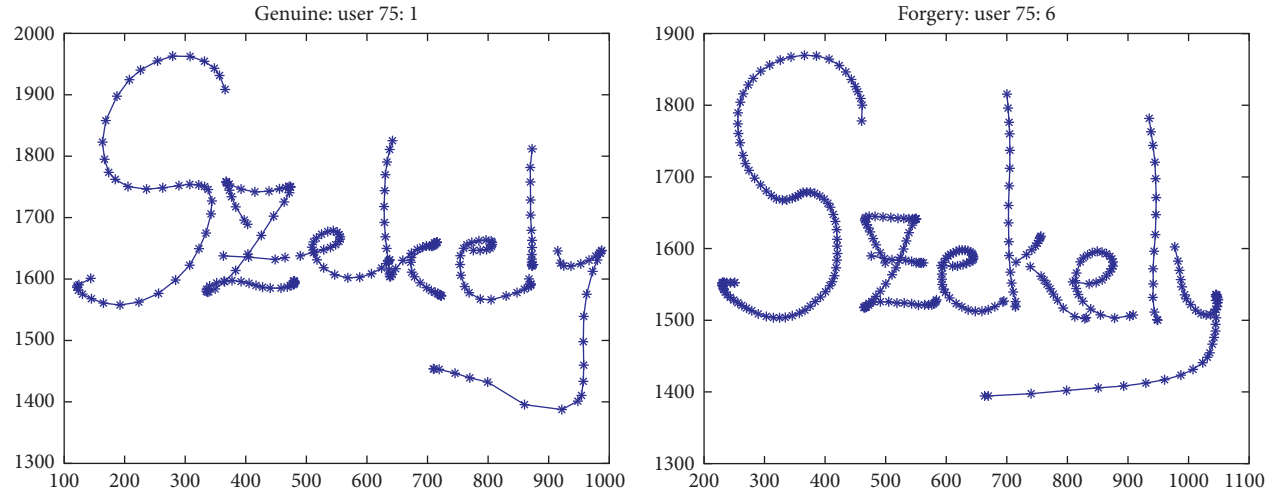
$$\text{Dscore} = \frac{1}{N} \sum_{i=1}^{N} d_{\text{DTW}}(\text{test}, e_i). \quad (1)$$

is transformed into a similarity score using (2):



(a)

Figure 2: Continued.



(b)

Figure 2: Pseudosignatures from the database. The first column contains genuine signatures and the second column contains forgeries.

$$\text{score} = \frac{1}{(1 + \text{Dscore})}. \quad (2)$$

The architecture of our function-based system is shown in Figure 3.

4.2. Feature-Based Verification. The second type of verification system is a feature-based or a global system, which computes a fixed size feature vector from each signature. Each feature is a global feature related to the signature as a whole. The components of our feature-based system are the following: feature extractor, user template creation, and matcher. No preprocessing was applied. However, before computations, features were scaled (separately for each user). The architecture of our feature-based system is shown in Figure 4.

Our feature extractor computed position- and time-based features [12, 22], such as duration and different types of velocity, as well as different types of sign change, a few sensor-related features (touchscreen pressure and finger area), and 2 types of histogram-based features. In the computation of histogram-based features, we followed the work of Sae-Bae and Memon [10] except that we used fewer features.

Let $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ be the x, y coordinates of a signature and $P = \{p_1, p_2, \ldots, p_n\}$ the pressure attribute. We compute the first- and second-order differences of these sequences as follows: $X^1 = \{x_i^1 \mid x_i^1 = x_{i+1} - x_i\}$, $Y^1 = \{y_i^1 \mid y_i^1 = y_{i+1} - y_i\}$, and $P^1 = \{p_i^1 \mid p_i^1 = p_i\}$, where $i = 1, 2, \ldots, n-1$, and $X^2 = \{x_i^2 \mid x_i^2 = x_{i+1}^1 - x_i^1\}$, $Y^2 = \{y_i^2 \mid y_i^2 = y_{i+1}^1 - y_i^1\}$, and $P^2 = \{p_i^2 \mid p_i^2 = p_{i+1}^1 - p_i^1\}$, $i = 1, 2, \ldots, n-2$.

Angles $\Theta_i^1 = \tan 2^{-1}(y_i^1, x_i^1), i = \overline{1, n-1}$, were computed in order to derive a uniform-width histogram from this sequence. The $\tan 2^{-1}$ trigonometric function is a common variation of the standard arctan function, which produces results in the range $(-\pi, \pi]$. Angles characterize the shape of the signature. This interval of angles was divided into 8 equal bins in order to compute the histogram.

```
(1) procedure DTWDist s[1···n], t[1···m]
(2)   DTW[0···n][0···m]
(3)   for i←1, n do
(4)     DTW[i][0]←infinity
(5)   end for
(6)   for i←1, m do
(7)     DTW[0][i]←infinity
(8)   end for
(9)   DTW[0][0]←0
(10)  for i←1, n do
(11)    for i←1, m do
(12)      cost←distance(s[i], t[j])
(13)      DTW[i][j]←cost + min(DTW[i−1][j]),
(14)      DTW[i][j−1], DTW([i−1][j−1] )
(15)    end for
(16)  end for
(17)  result←DTW[n][m]/(n + m)
(18) end procedure
```

Algorithm 1: DTW algorithm.

The $r_i^1 = \sqrt{(x_i^1)^2 + (y_i^1)^2}, i = \overline{1, n-1}$, sequence captures the speed distribution, and it is considered very useful in combating skilled forgeries [10]. In the histogram computation, we considered only the values from the interval $[0, \mu + 3\sigma]$, where μ represents the mean and σ the standard deviation of the r sequence obtained from a single signature. This interval was divided into 16 equal bins for the histogram computation. The full list of the features is presented in Table 2.

In feature-based methods, each signature is represented by a D-dimensional feature vector. After selecting the signatures used for template creation, features were scaled. We applied the normalization for each feature ($f_i' =$

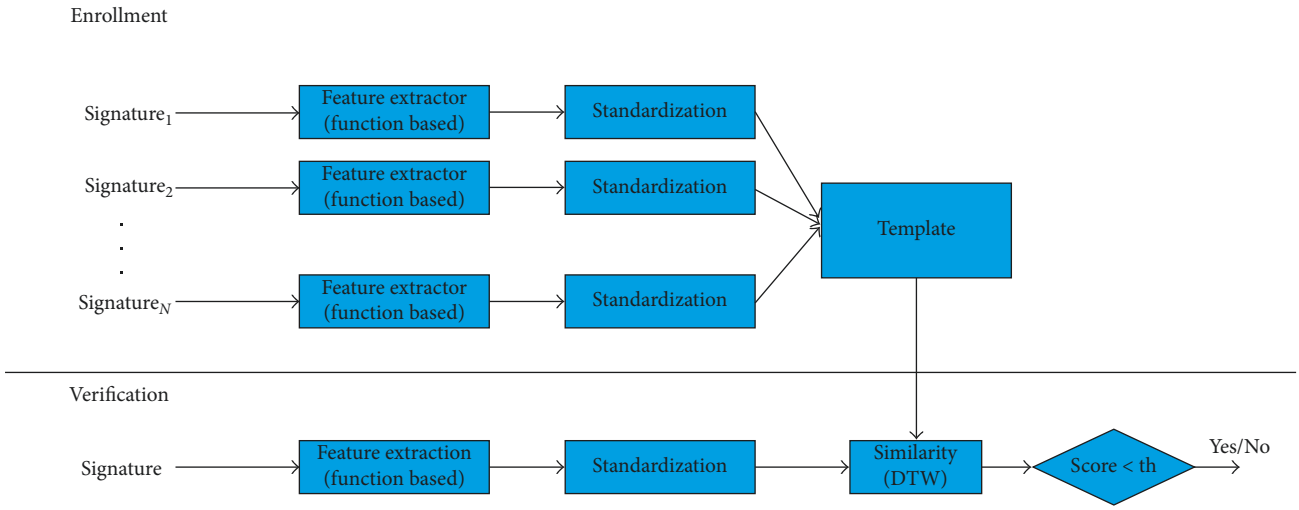


Figure 3: System architecture of function-based method.

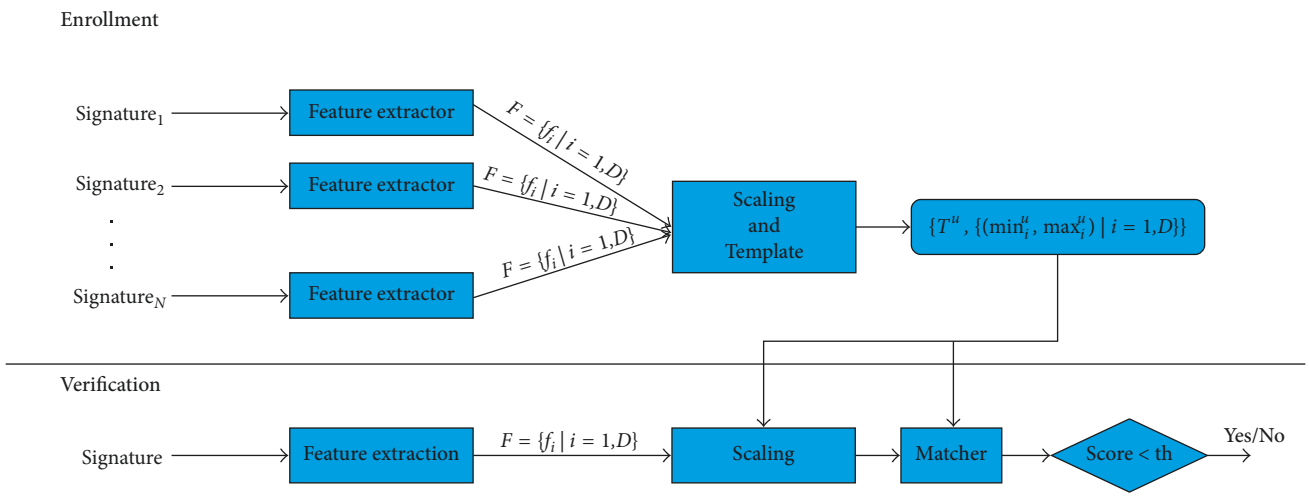


Figure 4: System architecture of feature-based method.

$(f_i - \min(f_i))/(\max(f_i) - \min(f_i)),\ i = 1, \ldots, D)$ before each template creation. In order to be able to apply the same scaling to the test signature before matching, the scaling parameters (max and min values for each feature) were stored as part of the user's template. User-based scaling of the data meets real biometric verification systems, because the system does not take into account the data of other users. The performance of the system is increased significantly by normalizing all users' data in a single step. This type of normalization was already evaluated for the DooDB database [23].

In data mining, anomaly detection (also outlier detection) is the identification of observations which do not conform to an expected pattern in a dataset [24] and therefore can be used for impostor detection. Killourhy and Maxion [25] evaluated 14 anomaly detectors for keystroke dynamic biometrics.

Four anomaly detectors were implemented for template creation and matching. Each anomaly detector uses a specific template and score computation (see below). Dissimilarity scores were transformed into similarity scores by using formula (2) as for the DTW algorithm.

In the training phase, user models (templates) are constructed using a fixed number of training samples. Testing or verification refers to an anomaly score computation for the test sample, which in our case is a distance from the user model.

The classic Euclidean anomaly-detection algorithm in the training phase calculates the mean vector of the training samples, while in the test phase it calculates the Euclidean distance between the mean vector and the test sample.

The Manhattan detector is similar to the Euclidean detector except that the Manhattan distance is used in the testing phase. Another variant of the Manhattan detector is the Manhattan scaled detector, described in Araujo et al. [26]. In the training phase, besides the mean vector of the training samples $(m_i, i = \overline{1, D})$, the mean absolute deviation of each feature is also calculated $(a_i, i = \overline{1, D})$. The anomaly

Table 2: Global features.

Description	#nf
Total signature duration	1
Number of finger strokes	1
Amount of time the finger touched the display	1
The number of sign changes of $x^1, y^1, x^2, y^2, p^1, p^2$	6
Mean of $x^1, y^1, x^2, y^2, p^1, fa$	6
Standard deviation of x^1, x^2, y^1, y^2	4
Maximum value of p^1	1
Mean velocity	1
Mean acceleration	1
Mean velocity in x direction	1
Mean velocity in y direction	1
Normalized point in time when the maximal value of $x, x^1, x^2, y, y^1, y^2, p^1$ is reached	7
Normalized point in time when the minimal value of x, x^1, x^2, y, y^1, y^2 is reached	6
Starting direction	1
Histogram of Θ sequence	8
Histogram of r sequence	16
Total	**62**

#nf, number of features.

Table 3: Number of testing samples used in evaluations.

Ev. type	Positive samples	Negative samples
SF	15 (session 2)	20
RF	15 (session 2)	1/each other user

SF, skilled forgeries; RF, random forgeries.

score is calculated using the formula: $\text{score}(x, \text{model}) = \sum_{i=1}^{D} |x_i - m_i|/a_i$, where x is a test sample $(x_i, i = \overline{1, D})$, and the model comprises the mean vector of the features and the mean absolute deviation of the features: $\text{model} = (m_i, a_i), i = \overline{1, D}$.

The k-nearest neighbour (kNN) detector works as follows: in the training phase, the detector saves the training samples (reference-based system). In the test phase, the detector calculates the Euclidean distance between each of the training samples and the test sample. The anomaly score is calculated as the mean of the distances to the k-nearest training samples.

One-class SVM is an outlier detector based on the support vector method. We used the LibSVM implementation [27] provided by e1071 package in R, with the following parameters: type = one-classification (for novelty detection), kernel = "radial," and gamma = 0.04. The SVM parameter nu was set to 0.4, and parameters were not tuned for individual users.

5. Benchmark Results

5.1. Evaluation Protocol. The same evaluation protocol was used for both verification systems, which consists of three measurements. The systems were trained with the first 5, 10, and 15 samples from the first session, respectively. The 15 genuine signatures from the second session were used for positive score computation. All of the 20 available forgeries per user are used for skilled forgery score computations. Random forgery scores were computed for each user by using the first genuine signature from all the other users (Table 3).

Two types of EERs were computed. The global EER was computed based on a global threshold, which was computed using common genuine and forgery score lists for all users of the database. The second type of EER, the a posteriori user-specific EER, is reported in some recent papers [13, 28]. This type of EER is computed by averaging the user-specific EERs. User-specific EERs are computed independently for each subject of the dataset and therefore are based on user-specific decision thresholds. Martinez-Diaz et al. [13] used the notation of aEER for this type of EER. As we compare our results with their results, it is important to use the same notation. However, it is important to note that this type of EER is based on a posteriori user-specific thresholds [29]. Hence, the corresponding EER (aEER) represents the best global EER that would be obtained if optimal score normalization techniques were known a priori [13].

5.2. Results

5.2.1. Training Set Size. The effect of the number of training samples during enrollment was investigated. Three cases were evaluated for each type of local and global features: using 5, 10, and 15 samples during enrollment.

Table 4 shows the function-based system results for the MOBISIG database. The five types of local feature sets were as follows: (i) xy—the coordinate sequence; (ii) x^1y^1—the first-order differences of the coordinate sequence; (iii) x^2y^2—the second-order differences of the coordinate sequence; (iv)$xyx^1y^1x^2y^2$—coordinate sequence with first- and second-order differences; (v) $xyx^1y^1x^2y^2pp^1$—coordinate sequence with first- and second-order differences and pressure with first-order differences. Both types of evaluations, skilled and random forgeries, are reported.

In the case of skilled forgeries, the best global EER was obtained by using the xy time sequences only (20.82%). However, the best aEER was 5.81% using type (v) local features.

In the case of random forgeries evaluation, the best result was obtained by using the first differences of x and y time sequences (EER: 1.41% and aEER: 0.01%). The very low EERs obtained show that this type of function-based verification system is highly reliable against random forgeries.

As expected, the more samples used in the enrollment phase, the better the verification system performance was. This is true for each type of local feature and for both types of evaluations: skilled and random forgeries. However, the improvements are small, especially between using 10 and 15 samples for enrollment.

The results obtained for the feature-based system are reported in the following. In order to show the contribution of different categories of global features, we formed three

Table 4: Verification performance in terms of EER and averaged individual EER (aEER) for the MOBISIG database using DTW and local features (%).

Features	Skilled forgeries		Random forgeries	
	EER	aEER (std dev.)	EER	aEER (std dev.)
Enrollment: 5 samples				
(i)	25.45	14.72 (14.62)	2.12	0.32 (1.29)
(ii)	31.37	11.42 (14.64)	1.75	0.00 (0.00)
(iii)	42.92	24.72 (21.46)	22.21	3.34 (9.27)
(iv)	32.02	10.28 (14.72)	3.18	0.10 (0.71)
(v)	31.52	8.56 (11.50)	4.51	0.18 (1.00)
Enrollment: 10 samples				
(i)	22.71	12.10 (13.98)	1.91	0.34 (1.31)
(ii)	28.87	8.60 (11.99)	1.41	0.01 (0.07)
(iii)	42.66	20.43 (18.34)	20.61	1.81 (3.52)
(iv)	30.35	6.80 (12.17)	2.55	0.01 (0.09)
(v)	29.97	6.69 (10.05)	4.09	0.16 (0.99)
Enrollment: 15 samples				
(i)	20.82	10.77 (13.29)	1.76	0.27 (1.18)
(ii)	27.15	7.34 (11.53)	1.68	0.01 (0.07)
(iii)	42.13	18.74 (17.49)	20.14	1.70 (3.60)
(iv)	29.79	6.27 (11.54)	2.62	0.01 (0.09)
(v)	29.76	5.81 (9.64)	4.27	0.15 (0.98)

feature sets from the features shown in Table 2: feat62 : all features; feat56 : feat62\{pressure and finger area features}; and feat32 : feat56\{histogram-based features}.

Table 5 shows the feature-based system results for the MOBISIG database using all features (feat62). As for the function-based system, we report results for using 5, 10, and 15 samples for enrollment. Similar to the function-based system, using more samples for enrollment improves the performance of the feature-based verification system. The performance gaps are higher for cases between 5 and 10 samples than for cases between 10 and 15 samples.

Comparing Table 5 with Table 4, it can be observed that the DTW system achieved far better performances in the case of random forgeries than the anomaly detectors. In the case of skilled forgeries, both the best global EER (14.31%) and the best aEER (9.35%) were obtained by the Manhattan detector. Interestingly, the differences between global EER and aEER are not so large as in the case of DTW system. The ROC curves for the global EERs are shown in Figure 5.

5.2.2. Global Feature Sets. The results obtained for the three global feature sets are shown in Table 6. The best (lowest) error rates were obtained by using the Manhattan detector both for skilled and random forgery evaluations. In the case of this detector, using less features resulted in higher error rates. However, not all detectors were affected negatively by using less features. For example, the SVM detector performance increased by reducing the dimension of the feature set, especially when the pressure-related features were excluded (feat56).

Table 5: Verification performance in terms of EER and averaged individual EER (aEER) for the MOBISIG database using anomaly detectors and 62 features (%).

Detector	Skilled forgeries		Random forgeries	
	EER	aEER (std dev.)	EER	aEER (std dev.)
Enrollment: 5 samples				
Eucl.	22.99	18.95 (15.81)	12.24	7.43 (9.13)
Manh.	19.27	13.57 (12.43)	7.28	3.60 (5.21)
kNN	21.68	17.19 (15.22)	11.85	6.96 (8.46)
SVM	29.81	18.34 (15.66)	25.78	7.61 (9.52)
Enrollment: 10 samples				
Eucl.	20.12	13.78 (12.84)	9.18	5.80 (8.99)
Manh.	16.58	10.82 (11.73)	5.37	2.60 (4.72)
kNN	18.40	13.35 (12.62)	8.63	5.80 (8.90)
SVM	19.93	14.03 (12.98)	10.59	5.80 (8.90)
Enrollment: 15 samples				
Eucl.	17.11	12.14 (12.60)	8.69	5.01 (8.47)
Manh.	14.31	9.35 (11.28)	5.21	2.10 (4.01)
kNN	17.25	12.15 (12.21)	9.14	5.00 (8.87)
SVM	17.35	12.45 (12.84)	9.00	5.00 (8.57)

5.2.3. Intersession Variability. We evaluated the intersession variability for the MOBISIG dataset. Two cases were evaluated: (1) using session 1 for training and session 2 for testing and (2) using session 2 for training and session 3 for testing. The results for using 15 samples for enrollment are shown in Table 7. The performance differences between the two cases are between 0.16% and 3.30% and are highly dependent on the used method. For example, verification results for system-wide EER are improved for the DTW (type (ii) features) method in case of session pair 2-3 (compared to session pair 1-2), but this tendency is not similar for aEER value in the case of the Manhattan detector.

According to the results from Table 7, no general tendency of improvement or deterioration of verification results could be stated. The results might suggest similar signature quality in session pairs 1-2 and 2-3 considering verification performance, but further investigations are necessary.

6. DooDB and MOBISIG comparison

In this section, comparative results are presented for our MOBISIG database and the other publicly available DooDB database. The results are presented along (i) their statistical properties, (ii) the verification system performances, (iii) score distributions, and (iv) some signature quality metrics.

6.1. Statistical Properties. Table 8 presents the most important characteristics regarding the data collection process of the two publicly available mobile device context signature databases.

6.2. Verification Performance. In the case of function-based DTW system, only the coordinates and their first- and second-

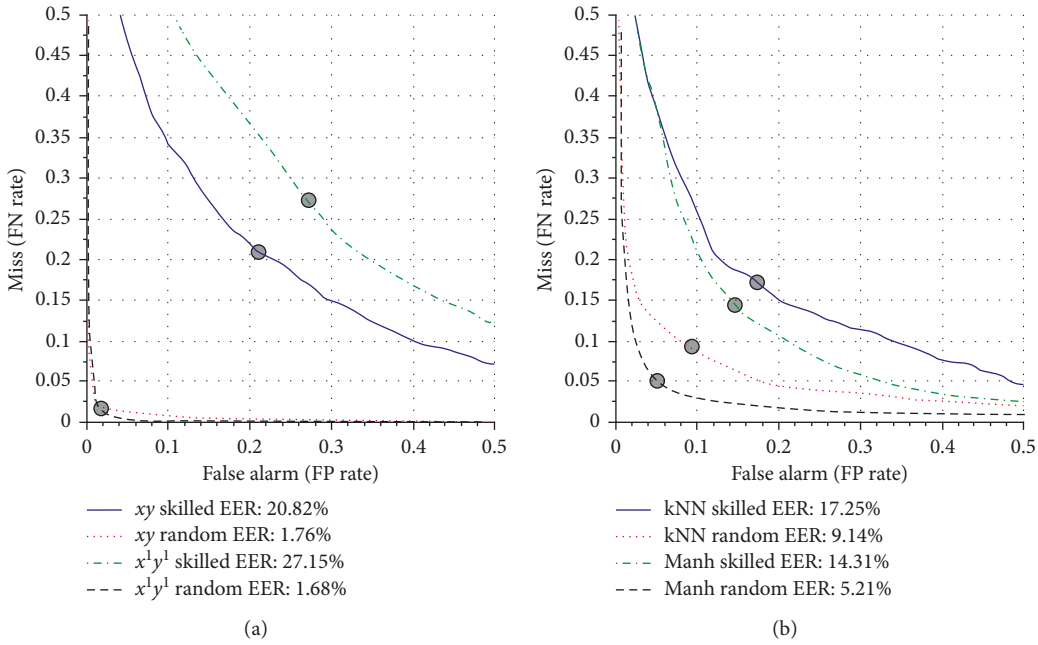


Figure 5: MOBISIG. ROC curves for local and global methods. 15 sample training cases. (a) DTW and (b) anomaly detectors.

Table 6: MOBISIG: EER and aEER (different feature sets).

Detector	Skilled forgeries		Random forgeries	
	EER	aEER (std dev.)	EER	aEER (std dev.)
		feat62		
Eucl.	17.11	12.14 (12.60)	8.69	5.01 (8.47)
Manh.	14.31	9.35 (11.28)	5.21	2.10 (4.01)
kNN	17.25	12.15 (12.21)	9.14	5.00 (8.87)
SVM	17.35	12.45 (12.84)	9.00	5.00 (8.57)
		feat56		
Eucl.	16.00	13.05 (13.55)	7.44	5.17 (8.64)
Manh.	15.45	10.97 (12.75)	5.26	2.32 (4.26)
kNN	16.91	13.48 (13.10)	7.46	5.56 (9.08)
SVM	15.73	13.22 (13.29)	7.39	5.35 (9.07)
		feat32		
Eucl.	17.58	14.87 (14.81)	8.80	5.89 (9.08)
Manh.	17.17	13.32 (13.62)	7.44	3.83 (6.56)
kNN	18.07	14.44 (13.62)	8.83	6.17 (9.09)
SVM	17.76	14.78 (14.86)	8.76	5.83 (9.15)

Note. Training: 15 samples.

Table 7: MOBISIG: EER and aEER (intersession variability).

Method	Skilled forgeries		Random forgeries	
	EER	aEER (std dev.)	EER	aEER (std dev.)
		Training: session 1; testing: session 2		
DTW	27.15	7.34 (11.53)	1.68	0.01 (0.07)
Manh.	14.31	9.35 (11.28)	5.21	2.10 (4.01)
SVM	17.35	12.45 (12.84)	9.00	5.00 (8.57)
		Training: session 2; testing: session 3		
DTW	24.69	9.05 (15.85)	1.78	0.33 (1.98)
Manh.	15.49	10.94 (12.27)	5.05	2.48 (4.33)
SVM	18.24	13.74 (13.66)	8.59	4.84 (6.52)

Note. Training samples: 15. We used type (ii) local features for DTW and 62 features (feat62) for the Manhattan anomaly detector and SVM.

order differences were used (no pressure information is available in the DooDB database). The pressure-based features were omitted for the feature-based system, consequently measurements were performed on feat56 and feat32 feature sets. The DooDB database contains $(0, 0)$ coordinate values in the case of sampling errors. In order to correct this error, coordinate values from the previous sample were assigned to the sample.

The same evaluation protocol was followed as presented in Section 5.1. We present the results obtained by the two verification methods on the DooDB database (Tables 9 and 10), followed by the comparison of the results obtained on the two databases (Table 11).

Comparing the skilled forgery results by the DTW method (Tables 4 and 9), we observe that using more samples for reference signatures decreases both global and user average EERs. Among the xy, x^1y^1, and x^2y^2 features, the first-order differences (x^1y^1) provide the lowest aEER for both databases (7.34% for MOBISIG and 16.67% for DooDB). However, combining all features ($xyx^1y^1x^2y^2$), the aEERs become slightly better (6.27% for MOBISIG and 15.78% for DooDB). Kholmatov and Yanikoglu [30] also reported that the first-order differences had given the lowest error rates. The worst results were obtained by using thesecond-order differences (x^2y^2) alone, which means that these features do not contain too much user-specific information.

Table 8: DooDB versus MOBISIG data collection.

	DooDB	MOBISIG
Device	HTC Touch DD mobile phone	Nexus 9 tablet
Touchscreen	Resistive	Capacitive
Frequency	100 Hz (periodic sampling)	Approx. 60 Hz (event-based sampling)
Coordinates	x: 0–2000, y: 0–3500	x: 0–1536, y: 0–2048
Raw data	$[x_t, y_t, \Delta_t]$	$[x_t, y_t, t, p_t, fa_t, vx_t, vy_t, ax_t, ay_t, az_t, gx_t, gy_t, gz_t]$
Sessions	2 (two weeks between)	3 (one week between)
Subjects	100 (44 women and 56 men)	83 (34 women and 49 men)
Samples	Session 1: 15 genuine + 10 forgeries Session 2: 15 genuine signatures + 10 forgeries	Session 1: 15 genuine signatures Session 2: 15 genuine signatures + 10 forgeries Session 3: 15 genuine signatures + 10 forgeries

Table 9: Verification performance in terms of EER and averaged individual EER (aEER) for the DooDB database using DTW (%).

Features	Skilled forgeries		Random forgeries	
	EER	aEER (std dev.)	EER	aEER (std dev.)
	Enrollment: 5 samples			
(i)	30.57	22.33 (19.60)	4.31	1.71 (8.27)
(ii)	31.93	19.63 (19.36)	4.96	1.37 (3.54)
(iii)	41.73	24.35 (20.19)	20.41	5.86 (9.28)
(iv)	32.15	19.68 (18.82)	4.44	1.14 (3.82)
	Enrollment: 10 samples			
(i)	29.48	19.59 (18.73)	3.63	1.51 (7.69)
(ii)	29.69	17.51 (18.35)	4.31	1.14 (2.55)
(iii)	42.25	21.49 (19.32)	19.63	5.05 (7.92)
(iv)	31.09	17.15 (17.74)	3.60	1.06 (3.78)
	Enrollment: 15 samples			
(i)	27.41	17.66 (17.57)	2.80	1.40 (7.66)
(ii)	29.27	16.67 (18.32)	3.71	1.12 (2.54)
(iii)	41.51	20.27 (19.02)	18.75	4.74 (7.69)
(iv)	30.12	15.78 (17.54)	2.94	1.15 (4.19)

Table 10: Verification performance in terms of EER and averaged individual EER (aEER) for the DooDB database using anomaly detectors and 56 features (%).

Detector	Skilled forgeries		Random forgeries	
	EER	aEER (std dev.)	EER	aEER (std dev.)
	Enrollment: 5 samples			
Eucl.	34.69	30.98 (17.91)	17.98	9.97 (11.95)
Manh.	31.24	25.29 (18.16)	13.38	5.11 (7.29)
kNN	35.33	31.42 (17.76)	18.58	10.29 (12.26)
SVM	35.6	31.73 (17.86)	27.74	10.84 (11.93)
	Enrollment: 10 samples			
Eucl.	27.632	24.63 (17.59)	12.20	7.61 (10.81)
Manh.	24.45	20.22 (16.49)	9.80	4.56 (7.08)
kNN	27.44	24.47 (17.74)	11.82	7.55 (10.33)
SVM	27.14	24.66 (17.38)	12.60	7.55 (10.79)
	Enrollment: 15 samples			
Eucl.	23.48	20.98 (16.05)	10.32	6.76 (10.28)
Manh.	21.71	18.75 (16.66)	7.82	4.19 (7.24)
kNN	24.25	20.83 (16.19)	10.10	7.07 (10.43)
SVM	23.26	20.85 (16.07)	10.40	6.97 (10.39)

The effect of using a reduced feature set is shown in Table 12. It can be seen that not using the histogram-based features resulted in a very small performance degradation (about 1%).

The ROC curves for some verification method are shown in Figure 6.

6.3. Score Distributions. The score distributions of genuine and impostor samples (skilled forgeries case, 15 training samples) for both verification methods are shown in Figure 7.

6.4. Signature Quality. Alonso-Fernandez et al. [31] reviewed the state of the art in the biometric quality problem. They consider that a biometric sample is of good quality if it is suitable for personal recognition. According to the ISO/IEC 29794-1 standard, biometric quality has three dimensions: fidelity, utility, and character. Fidelity is the degree of similarity between the sample and its source. Utility is related to the impact of the sample on the overall performance of a biometric system, while character indicates the inherent discriminative capability of the source.

Influence of the character of a biometric sample on its utility was investigated by Müller and Henniger [15]. In case of signatures, recognition systems will give different results depending on the reference samples (the sample set used for template creation). Selection of samples used in the template can be controlled by sample quality assessment algorithms. Evaluating the quality of a single sample based only on sample features is difficult, but methods exist to evaluate considering other samples. Some a posteriori methods use only genuine samples, others use skilled forgery samples also. Using a set of genuine samples for quality assessment still restricts selection of reference samples to the genuine user and assures usage of the method in real applications. In consequence, we used two metrics which examine the character of the sample proposed by Müller and Henniger [15].

The first method, the sampleEER, consists of computing the EER for each genuine sample. This is obtained by

Table 11: EER and aEER comparison for DooDB and MOBISIG databases.

Method	Skilled forgeries		Random forgeries	
	EER	aEER	EER	aEER
		DooDB		
Manh.	21.71	18.75 (16.66)	7.82	4.19 (7.24)
DTW	29.27	16.67 (18.32)	3.71	1.12 (2.54)
		MOBISIG		
Manh.	15.45	10.97 (12.75)	5.26	2.32 (4.26)
DTW	27.15	7.34 (11.53)	1.68	0.01 (0.07)

Note. 15 training samples (session 1); type (ii) local features for DTW and 56 features for the Manhattan anomaly detector (%).

Table 12: Verification performance in terms of EER and averaged individual EER (aEER) for the DooDB database using anomaly detectors (different feature sets).

Detector	Skilled forgeries		Random forgeries	
	EER	aEER (std dev.)	EER	aEER (std dev.)
		feat56		
Eucl.	23.48	20.98 (16.05)	10.32	6.76 (10.28)
Manh.	21.71	18.75 (16.66)	7.82	4.19 (7.24)
kNN	24.25	20.83 (16.19)	10.10	7.07 (10.43)
SVM	23.26	20.85 (16.07)	10.40	6.97 (10.39)
		feat32		
Eucl.	24.66	22.59 (17.09)	11.20	7.79 (11.00)
Manh.	22.60	19.63 (16.10)	8.64	5.19 (8.20)
kNN	25.43	22.68 (16.81)	11.34	8.21 (10.74)
SVM	24.60	22.43 (17.05)	11.55	7.88 (11.05)

Note. Training: 15 samples.

computing the distances to all the other corresponding genuine and forgery samples. We formed a positive and a negative score lists from the obtained distances (scores) and computed the EER. We computed DTW distances between signatures using type (i) local features.

The second metric, the userAvgDist, was computed as the average of the pairwise distances (DTW distance—type (i) local features) of genuine samples for each user (in the case of having N genuine samples, we computed $N(N-1)/2$ distances). The lower it was, the more similar the samples of the user were, so the more stable the user's signature was. For both databases, we used the first $N = 30$ genuine samples for computations.

After applying the two metrics for each genuine sample (sampleEER) or each user userAvgDist, we obtained two sequences of values. In order to characterize the dataset, both the mean and the standard deviation of each sequence were computed. The obtained results are shown in Table 13.

From the two quality metrics proposed by Müller and Henniger, we favor the userAvgDist because this metric does not rely on forgery samples; therefore, it could be used during data collection. Samples that are not closely similar to those already collected should be discarded.

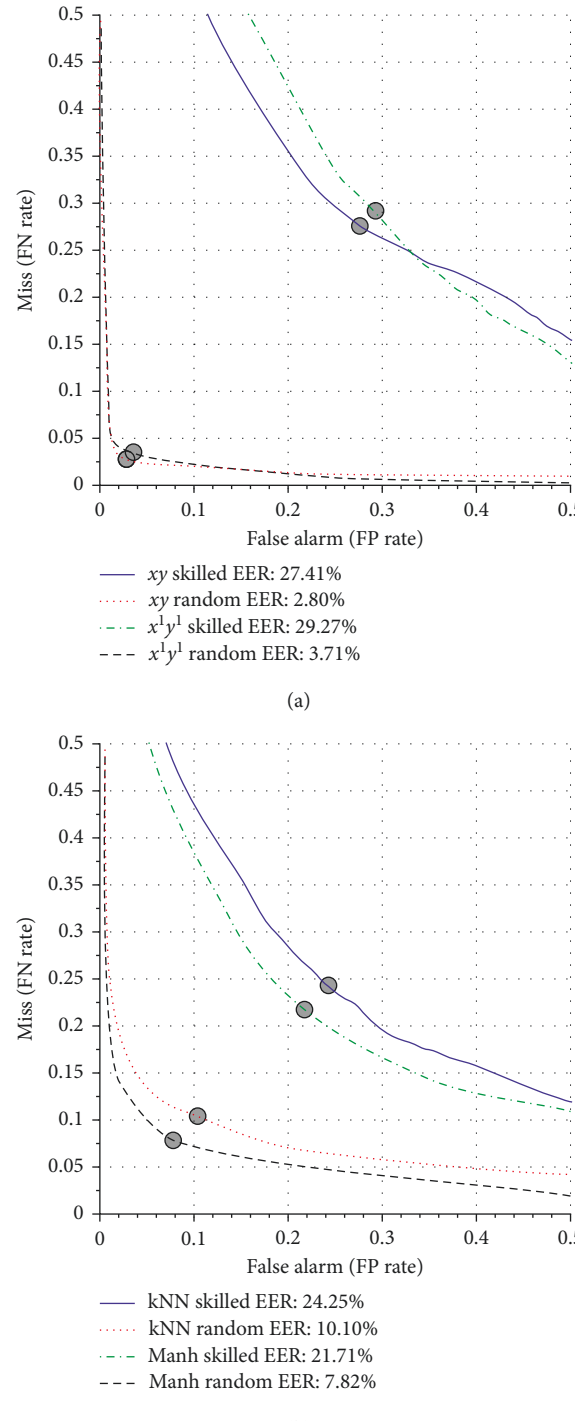


Figure 6: DooDB. ROC curves for local and global methods. 15 sample training cases. (a) DTW and (b) anomaly detectors.

7. Conclusion

In this paper, the MOBISIG finger-drawn online signature dataset has been presented. The dataset comprises pseudo-signatures from 83 users, both genuine and forgery samples. Benchmark verification experiments have been performed

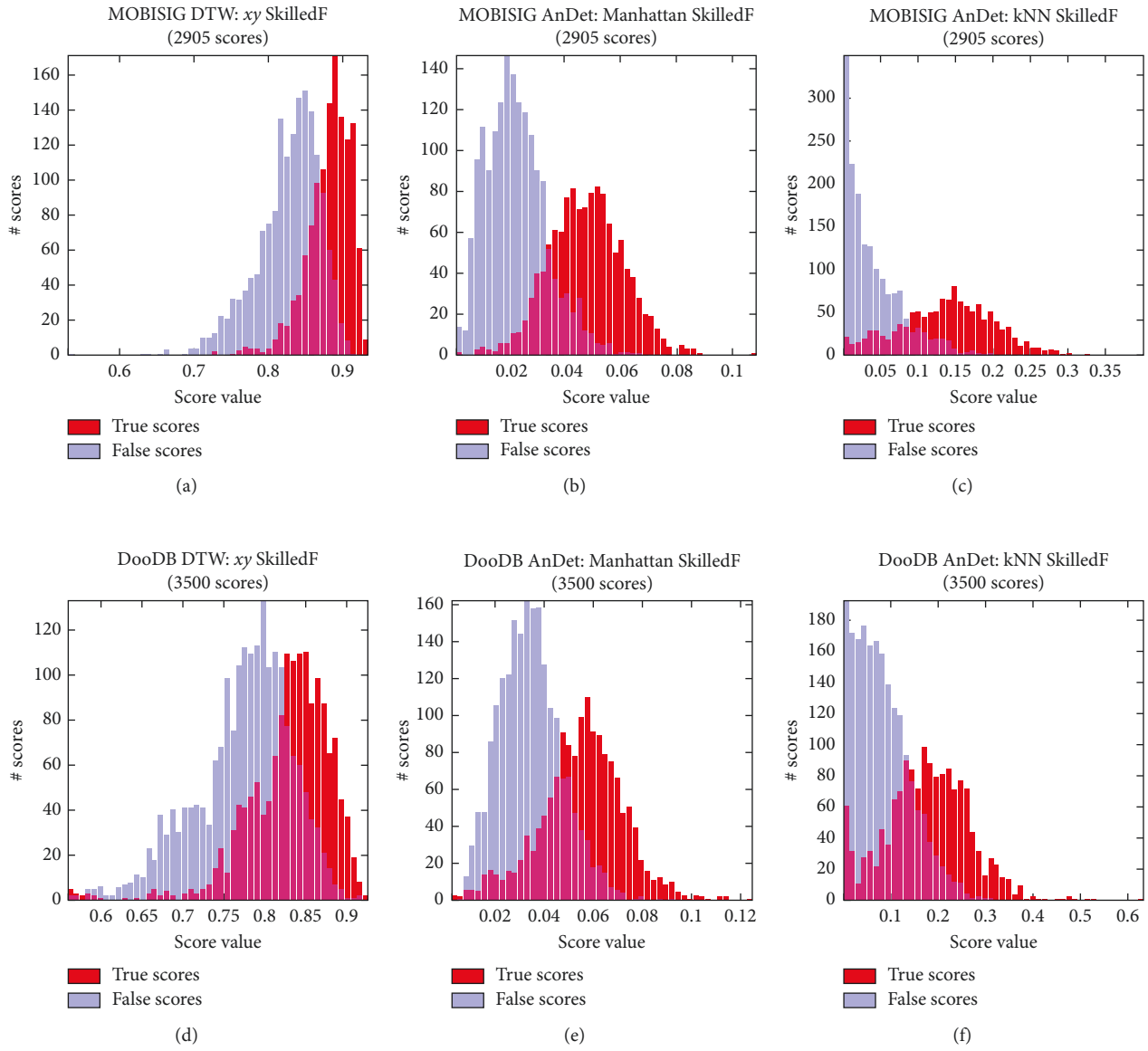


Figure 7: Score distributions. Training: 15 samples (session 1). Testing: genuine signatures (15 samples (session 2)) and skilled forgeries (20 samples). (a) MOBISIG-DTW-xy, (b) MOBISIG-Manhattan detector, (c) MOBISIG-kNN detector, (d) DooDB-DTW-xy, (e) DooDB-Manhattan detector, and (f) DooDB-kNN detector.

using both function and feature-based methods. Two types of EERs are reported: one is based on global threshold, while the other one on a posteriori user-specific thresholds. However, the global threshold-based EER results are not outstanding. Nevertheless, further improvement may be obtained by score normalization techniques. Good results were obtained using a posteriori user-specific thresholds (aEER). The lowest aEER for the skilled forgery case was 5.81% and 0.01% for the random forgery case. Both results were obtained by the function-based DTW method. Although feature-based methods offer poor results in the case of global threshold, they are significantly better than using function-based methods (skilled forgery case). The lowest aEERs were 9.35% (skilled forgery) and 2.10% (random forgery), both obtained by the Manhattan outlier detector.

Table 13: Signature quality measures for MOBISIG and DooDB databases (%).

Quality method	MOBISIG	DooDB
sampleEER	14.94 (12.58)	16.84 (12.59)
userAvgDist	12.85 (1.57)	19.85 (3.45)

The second objective of this paper was to compare our new dataset with similar publicly available ones. The only publicly available dataset collected on mobile devices and containing finger-drawn signatures was the DooDB. Therefore, we have presented a comparison of our MOBISIG and the DooDB dataset along (i) their statistical properties, (ii) the verification system performances using exactly the same methods and features, and (iii) some signature quality metrics. Signature

quality measures indicate slightly better values in the case of the MOBISIG dataset. This difference is also expressed by better results in all evaluations in this study for the MOBISIG dataset compared to the DooDB. These results might be explained by the larger touchscreen for data acquisition in the case of the MOBISIG dataset, furthermore by the average duration of individual signatures in MOBISIG dataset (twice as long as those in the DooDB dataset).

Future work may include the introduction of new signature quality metrics for online signatures, as well as their evaluation on several datasets.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research has been supported by Sapientia Foundation–Institute for Scientific Research.

References

[1] D. Impedovo and G. Pirlo, "Automatic signature verification: the state of the art," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 38, no. 5, pp. 609–635, 2008.

[2] J. G. A. Dolfing, E. H. L. Aarts, and J. J. G. M. van Oosterhout, "On-line signature verification with hidden Markov models," in *Proceedings of the Fourteenth International Conference on Pattern Recognition*, vol. 2, pp. 1309–1312, Brisbane, Queensland, Australia, August 1998.

[3] D. Y. Yeung, H. Chang, Y. Xiong et al., *SVC2004: First International Signature Verification Competition*, Springer Berlin Heidelberg, Berlin, Germany, 2004.

[4] A. Kholmatov and B. Yanikoglu, "SUSIG: an on-line signature database, associated protocols and benchmark results," *Pattern Analysis and Applications*, vol. 12, no. 3, pp. 227–236, 2009.

[5] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon et al., "MCYT baseline corpus: a bimodal biometric database," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 150, no. 6, pp. 395–401, 2003.

[6] S. Garcia-Salicetti, C. Beumier, G. Chollet et al., *BIOMET: A Multimodal Person Authentication Database Including Face, Voice, Fingerprint, Hand and Signature Modalities*, Springer Berlin Heidelberg, Berlin, Germany, 2003.

[7] J. Fierrez, J. Galbally, J. Ortega-Garcia et al., "BiosecurID: a multimodal biometric database," *Pattern Analysis and Applications*, vol. 13, no. 2, pp. 235–246, 2010.

[8] N. Houmani, S. Garcia-Salicetti, B. Dorizzi, and M. El-Yacoubi, *On-Line Signature Verification on a Mobile Platform*, Springer Berlin Heidelberg, Berlin, Germany, 2012.

[9] R. P. Krish, J. Fierrez, J. Galbally, and M. Martinez-Diaz, *Dynamic Signature Verification on Smart Phones*, Springer Berlin Heidelberg, Berlin, Germany, 2013.

[10] N. Sae-Bae and N. Memon, "Online signature verification on mobile devices," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 933–947, 2014.

[11] M. Martinez-Diaz, J. Fierrez, and J. Galbally, "The DooDB graphical password database: data analysis and benchmark results," *IEEE Access*, vol. 1, pp. 596–605, 2013.

[12] P. Bissig, "Signature verification on finger operated touchscreen devices," Master's thesis, ETH Zürich, Distributed Computer Group, Zürich, Switzerland, 2011.

[13] M. Martinez-Diaz, J. Fierrez, and J. Galbally, "Graphical password-based user authentication with free-form doodles," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 4, pp. 607–614, 2016.

[14] N. Houmani, A. Mayoue, S. Garcia-Salicett et al., "Biosecure signature evaluation campaign (BSEC'2009): evaluating online signature algorithms depending on the quality of signatures," *Pattern Recognition*, vol. 45, no. 3, pp. 993–1003, 2012.

[15] S. Müller and O. Henniger, *Evaluating the Biometric Sample Quality of Handwritten Signatures*, Springer Berlin Heidelberg, Berlin, Germany, 2007.

[16] N. Houmani, S. Garcia-Salicetti, and B. Dorizzi, "A novel personal entropy measure confronted with online signature verification systems' performance," in *Proceedings of the 2008 2nd IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS 2008)*, pp. 1–6, Washington, DC, USA, October 2008.

[17] N. Houmani, S. Garcia-Salicetti, and B. Dorizzi, "On measuring forgery quality in online signatures," *Pattern Recognition*, vol. 45, no. 3, pp. 1004–1018, 2012.

[18] N. Houmani and S. Garcia-Salicetti, "On hunting animals of the biometric menagerie for online signature," *PLoS One*, vol. 11, no. 4, article e0151691, pp. 1–26, 2016.

[19] C. Kahindo, S. Garcia-Salicetti, and N. Houmani, "A signature complexity measure to select reference signatures for online signature verification," in *Proceedings of the 2015 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–8, Darmstadt, Germany, September 2015.

[20] R. Guest and O. Henniger, "Assessment of the quality of handwritten signatures based on multiple correlations," in *Proceedings of the 2013 International Conference on Biometrics (ICB)*, pp. 1–6, Madrid, Spain, June 2013.

[21] E. N. Zois, L. Alewijnse, and G. Economou, "Offline signature verification and quality characterization using poset-oriented grid features," *Pattern Recognition*, vol. 54, pp. 162–177, 2016.

[22] J. Fierrez-Aguilar, L. Nanni, J. Lopez-Peñalba, J. Ortega-Garcia, and D. Maltoni, "An on-line signature verification system based on fusion of local and global information," in *Audio- and Video-Based Biometric Person Authentication, vol. 3546 of Lecture Notes in Computer Science*, pp. 523–532, Springer Berlin Heidelberg, Berlin, Germany, 2005.

[23] M. Antal and L. Z. Szabó, "On-line verification of finger drawn signatures," in *Proceedings of the 2016 IEEE 11th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 419–424, Timisoara, Romania, May 2016.

[24] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[25] K. Killourhy and R. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *Proceedings of the IEEE/IFIP International Conference on Dependable Systems Networks (DSN'09)*, pp. 125–134, Lisbon, Portugal, June–July 2009.

[26] L. C. F. Araujo, L. H. R. Sucupira, M. G. Lizarraga, L. L. Ling, and J. B. T. Yabu-Uti, "User authentication through typing biometrics features," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 851–855, 2005.

[27] C. C. Chang and C. J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[28] A. Morales, J. Fierrez, R. Tolosana et al., "Keystroke biometrics ongoing competition," *IEEE Access*, vol. 4, pp. 7736–7746, 2016.

[29] J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez, "Target dependent score normalization techniques and their application to signature verification," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 35, no. 3, pp. 418–425, 2005.

[30] A. Kholmatov and B. Yanikoglu, "Identity authentication using improved online signature verification method," *Pattern Recognition Letters*, vol. 26, no. 15, pp. 2400–2408, 2005.

[31] F. Alonso-Fernandez, J. Fierrez, and J. Ortega-Garcia, "Quality measures in biometric systems," *IEEE Security and Privacy Magazine*, vol. 10, no. 6, pp. 52–62, 2012.