Received: 10 April 2020 | Revised: 29 July 2020 | Accepted: 9 September 2020

DOI: 10.1049/bme2.12003

REVIEW

The Institution of Engineering and Technology WILEY

Mobile keystroke dynamics for biometric recognition: An overview

Emanuele Maiorana | **Himanka Kalita** | **Patrizio Campisi**

Department of Engineering, Roma Tre University, Roma, Italy

Correspondence

Emanuele Maiorana, Department of Engineering, Roma Tre University, Via Vito Volterra 62, 00146, Roma, Italy.
Email: emanuele.maiorana@uniroma3.it

Abstract

In the relatively recent past, the analysis of keystroke dynamics for biometric recognition purposes has intrigued researchers, since practical evidences have shown differences in the typing behaviours of distinct subjects. This area of research has become even more appealing since the emergence and evolution of mobile smartphones, given their pervasiveness and intensive use in real-life applications. In addition, unlike hard keyboards used with computers, mobile smartphones offer the possibility of exploiting embedded sensors to augment the information acquired when typing on their soft keyboards. This study discusses the state of the art of keystroke-dynamics-based automatic recognition system, exclusively when dealing with mobile devices, for both verification and identification purposes. In more details, the databases employed, the features selected, the methodologies implemented, and the performance achieved through the technological advances introduced in the last years, are here overviewed.

1 | INTRODUCTION

In the last decades, biometric technology has emerged, at an increasing rate, as an enabling solution for automatic people recognition, thanks to several intrinsic advantages it offers over conventional approaches. In fact, differently from traditional recognition methods, which exploit either what a person knows, like a password, or what a person owns, such a token, biometric recognition systems rely on who a person is, what a person does, or how a person answers to specific external stimuli [1]. In layman terms, biometric recognition systems perform pattern recognition tasks, in the form of either verification or identification, leveraging on features extracted from either physiological characteristics such as fingerprint, iris, face, and vein patterns, behavioural traits like voice, signature, gait, and keystroke, or even cognitive properties derived from responses of the nervous system such as those collected through electroencephalography or electrocardiography, to cite some examples. With respect to conventional approaches for automatic people recognition relying on passwords or tokens, biometric recognition systems exploit identifiers that, to some extent, cannot be lost, forgotten, stolen, copied, or forged [2]. Whereas in the past biometric recognition has been mostly performed on desktop devices for commercial applications [3, 4], in recent years we are witnessing an increasing use of this technology on handheld devices [5].

Mobile communication is, in fact, widespread, as shown in Figure 1 where the number of mobile subscriptions from 1993 to 2019 is reported. As can be seen, the current number exceeds the world population. Among the reasons for such astonishing growth from the early years of mobile communications to the present day, there is the fact that a huge number of people around the world use their cellular subscriptions to access the Internet, especially since the introduction of smartphones. Figure 2 shows that the number of worldwide smartphone users has raised at a very high rate, from 2.5 billion in 2016 to an expected 3.8 billion in 2021, out of an overall 5.8 billion users of mobile devices. A further growth, up to five billion smartphone users, is expected by 2025 [6]. Figure 3 further details the expected behaviour of worldwide mobile internet penetration, going beyond 60% by 2025, while remaining below 50% only in Sub-Saharan Africa.

Thanks to their portability and ease of use in comparison to bulky desktop systems, and also to the availability of fast wireless internet connections, more and more complex online tasks are nowadays performed through mobile devices, such as online banking, e-commerce, and so forth. As shown in Figure 4, the share of internet users employing mobile online

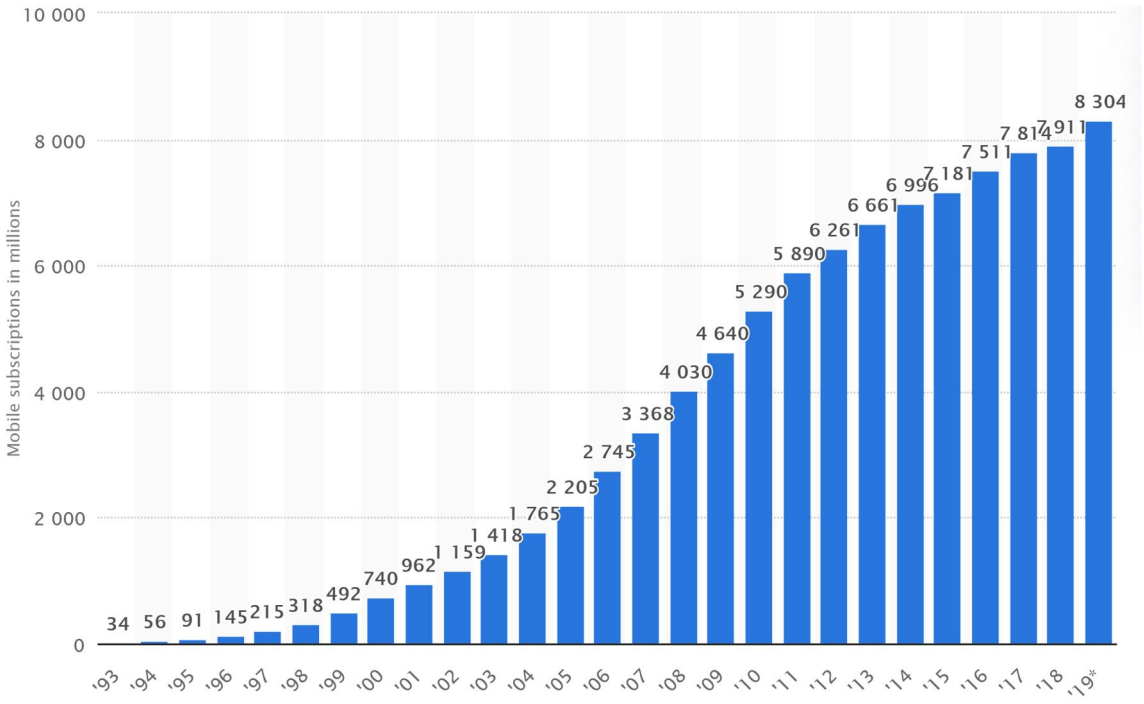

FIGURE 1 ITU (5 November 2019). Number of mobile (cellular) subscriptions worldwide from 1993 to 2019 (in millions; Graph). In Statista. Retrieved 20 May 2020, from https://www.statista.com/statistics/262950/global-mobile-subscriptions-since-1993/. ITU, International telecommunication union

payment services is in fact above 30% in most regions of the world.

Performing such tasks typically requires the provision, through the employed mobile devices, of sensitive and valuable data, such as personal identifiers, passwords, bank accounts, credit card numbers, booking orders, and so forth. Consequently, severe issues regarding the security and privacy of such data arise. In fact, there is either the possibility

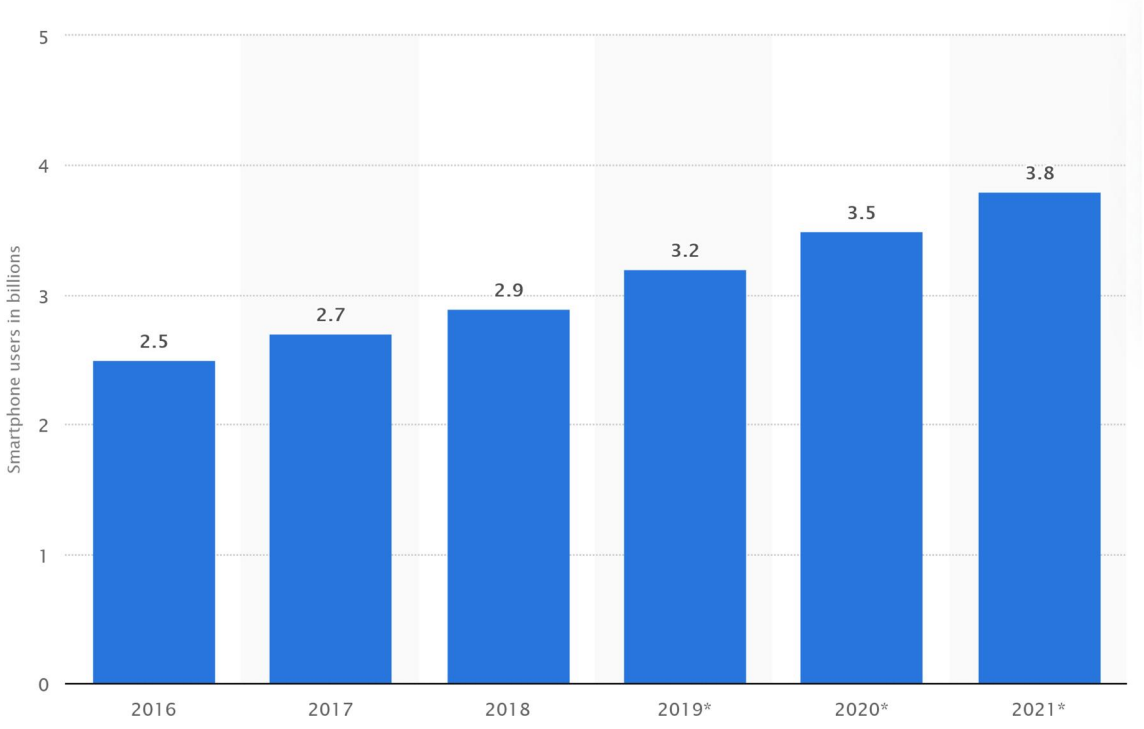

FIGURE 2 Newzoo (17 September 2019). Number of smartphone users worldwide from 2016 to 2021 (in billions; Graph). In Statista. Retrieved 1 July 2020, from https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

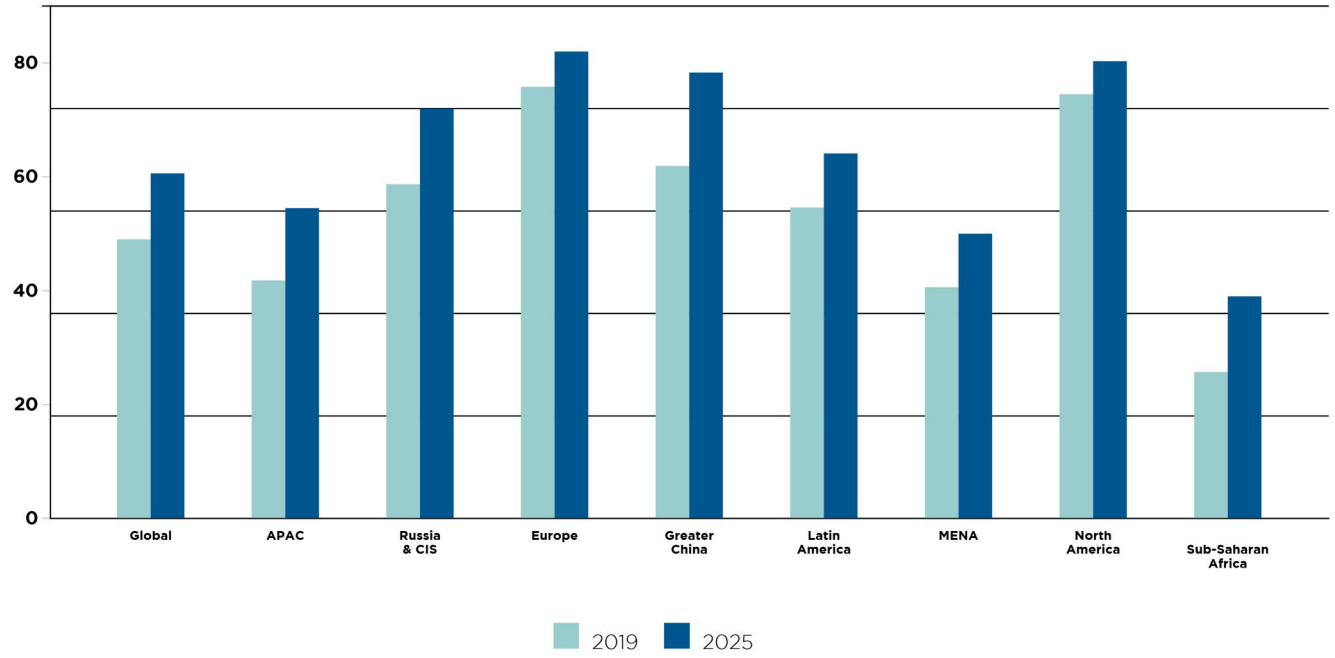

FIGURE 3 GSMA Intelligence (3 March 2020). Mobile internet penetration in 2020 and expectations for 2025 (APAC, Asia-Pacific; MENA, Middle-East and North Africa). Retrieved 1 July 2020, from https://www.gsma.com/mobileeconomy/. GSMA, global system for mobile communications association

that an unauthorized attacker could get access to them, or the possibility that people sharing our personal space, like children, could get access to unsuitable content [7]. The demand for secure methods to be used on mobile devices access control is therefore very high. For this reason, more and more smartphones embed biometric sensors to perform fingerprint-, face-, or iris-based recognition. At the same time, research on additional biometric recognition methods for mobile devices is also being carried out [8]. Among the performed activities, notable interest is devoted to the development of solutions which would not require the usage of any dedicated hardware, to contain the production costs.

Within this scenario, resorting to keystroke dynamics (KD) as biometric identifier on mobile devices seems a natural choice. KD is a behavioural characteristic that can be used to obtain discriminative information about a user by evaluating his/her typing capabilities. A first understanding about the individual nature of KD was gained by telegraphists in the mid 1800's, where operators working on a telegraph were recognized by their tapping skills [9], which were actually quite different from operator to operator. The first experimental tests confirming the existence of discriminative traits within the typing behaviour of each subject date back to 1980 [10]. KD can be acquired without asking the users to perform any specific action. Furthermore, unlike the majority of other biometric traits, KD allows

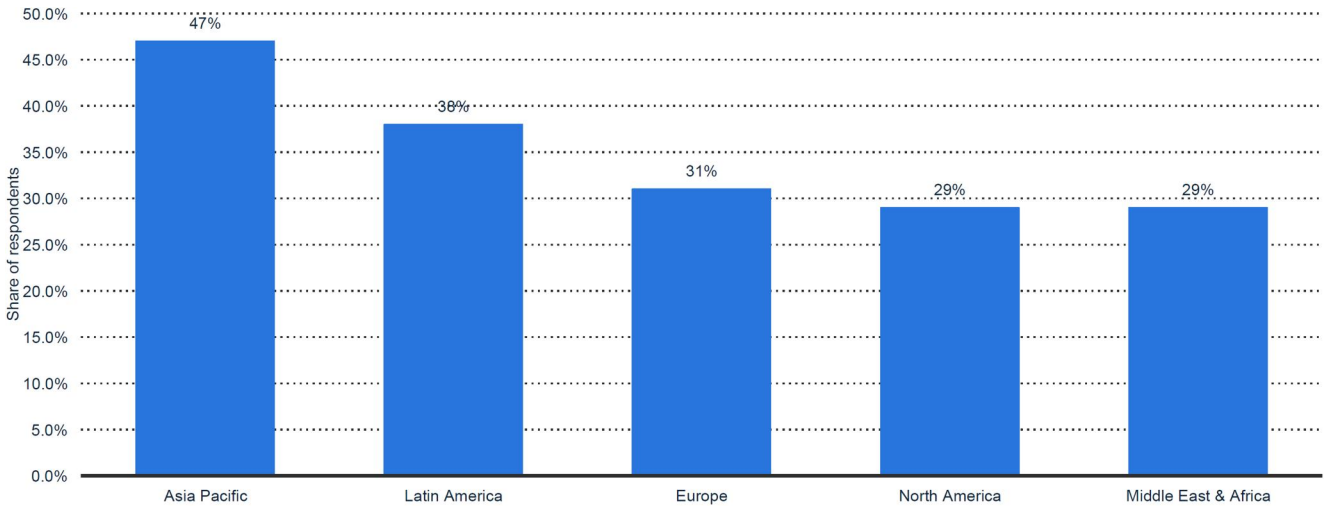

FIGURE 4 Share of internet users employing mobile online payment services as of fourth quarter of 2018, by region. In Statista. Retrieved 1 July 2020, from https://www.statista.com/study/21391/mobile-internet-usage-statista-dossier/

performing both static and continuous recognition, with the latter modality providing robustness against session hi-jacking, in which an intruder may seize control of an ongoing session after a successful login of a legitimate user [11]. As already mentioned, recording KD also does not require the usage of any additional hardware, since the habitual typing rhythm of a person can be collected by simply using a keystroke logging software, capturing the timings associated with key-related interactions. Moreover, modern smartphones represent a particularly suitable environment to perform KD-based recognition, due to the availability of several embedded sensors, not necessarily engineered for biometric recognition purposes [12]. As a matter of fact, the gyroscope included in all modern mobile devices may provide information about the orientation of a phone, its motion, or which hand is used to hold it. Through the accelerometer, it could be possible to estimate the user's speed of walking. The global positioning system sensor could provide information about the location of the device. Touch sensors could also provide data about the way a user touches the phone and swipes on the screen [13].

The increasing relevance on this KD-based recognition is also witnessed by the recent interest of banks and retailers in analysing how a user types on the smartphone screen, and how a phone is held while performing an online transaction, to verify the customers' identity[1]. The interaction with text-entry interfaces has been also recently recognized by the European Banking Authority (EBA) as a reliable recognition solution for multi-factor authentication [14].

This study surveys the state of the art of KD-based automatic recognition systems developed for mobile devices. The first investigation about the feasibility of recognizing people through their typing behaviour on mobile devices has been reported in [15]. Such early attempt has been performed using a modified mobile Nokia 5110 phone, removing all parts from the handset with the exception of the keypad interface, which was connected to a desktop. Since then, several works have analysed the discriminative capability of KD on mobile devices. The temporal distribution of publications on this specific topic is reported in Figure 5.

This study is organized as follows: the signals which can be acquired by commercial mobile devices, and employed for KD-based biometric recognition, are detailed in Section 2. Both 'classical' characteristics such as those based on timing information, and 'advanced' traits related to the exploitation of embedded sensors such as gyroscope, accelerometer, and so forth [16], are analysed. Section 3 outlines the datasets, either public or private, collected in state-of-the-art approaches dealing with KD-based biometric recognition on mobile phones. The features exploited in the studies covered in this overview are detailed in Section 4, while the classification approaches proposed for accomplishing either identification or verification are discussed in Section 5. A comparative analysis of the obtained recognition performance, together with a summary of the findings derived from the analyses performed in literature, is given in Section 6. Section 7 outlines future research directions, while conclusions are finally drawn in Section 8.

[1]https://www.nytimes.com/2018/08/13/business/behavioral-biometrics-banks-security.html

2 | MOBILE KEYSTROKE DYNAMICS: AVAILABLE DATA

When dealing with KD, both in the desktop and mobile framework, different scenarios can be considered to acquire keystroke-related data [17]:

- Fixed text, which includes:
  - Personal identification numbers (PINs), that is, sequences of numbers used, for example, to unlock mobile subscriber identity modules (SIMs) or to withdraw cash from Automated teller machines. The length of PINs used to access mobile phones typically consists of four digits, yet numerical sequences with a higher number of digits could be also employed, depending on the considered applications and the required security level;
  - Usernames or passwords, consisting of sequences of alphabetic or alphanumeric characters. The common password length ranges from six to 15 characters, including also special characters, usually employed to increase security;
  - Passphrases, with users typing sentences or paragraphs pre-set during the enrolment stage, then replicated during the recognition stage. In this case, the length of the provided inputs may be in the order of several dozens of characters;
- Free text, where users, different from the previous cases, are free to provide any input, regardless what has been previously recorded during the enrolment phase. One-time passwords (OTPs) can be considered as a special case of free-text input, since they can be used only in one occasion, after which the sequence can no longer be repeated and used.

While subjects provide their inputs, different signals can be recorded and employed to characterize the users. In the following, we detail the information which can be made easily available by smartphones working under the Android OS. Specifically, timing information is described in details in Section 2.1, while the signals recorded by standard sensors embedded in modern mobile devices are outlined in Section 2.2. A summary is given in Table 1.

2.1 | Timing information

Most of the approaches exploiting KD for biometric recognition purposes rely on the timing information to discriminate among users. Mobile devices usually offer the availability of timestamp information, recording the instants when different keys are pressed and released. From such basic information, several latency measures can be then computed.

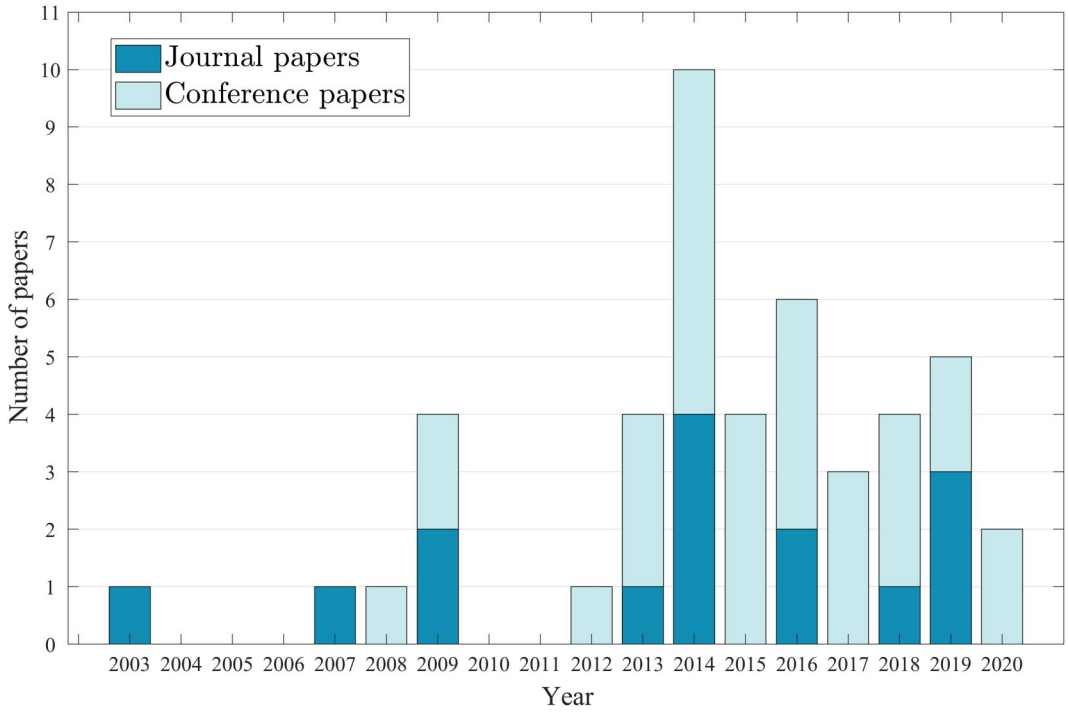

FIGURE 5 Papers dealing with KD-based biometric recognition on mobile devices; KD, keystroke dynamics

2.1.1 | Timestamp

The traditional time-based signals extracted from KD data record the key-press and key-release events as follows:

- Key-press timestamp (*KPT*), which is the system timestamp value, at the moment of key press, typically recorded in milliseconds;
- Key-release timestamp (*KRT*), which is also the system timestamp value, at the moment of key release, in milliseconds.

From the aforementioned raw information, several other discriminative measures can be derived.

2.1.2 | Latency

Several distinct measures of latency can be generated from the recorded timestamps. The most-commonly used information regards the key hold/dwell time (*KHT* or *KDT*), representing the time taken from pressing a key to its release. For a given *n*-th pressed key, KHT_n is evaluated as $KHT_n = KRT_n - KPT_n$. With reference to Figure 6, the subscripts refer to the order of key press and release events.

More in general, given an input string and any two distinct characters within it, each associated to one key, the following latencies can be considered [18]:

- Key press-release time (*PRT*), corresponding to the lapse from the press of a key to the release of the other one. This is often referred to as generalized key dwell time;
- Key press-press time (*PPT*), corresponding to the lapse from the press of a key to the press of the other one;
- Key release-press time (*RPT*), corresponding to the lapse from the release of a key to the press of the other one. This is often referred to as key flight time (*KFT*);
- key release-release time (*RRT*), corresponding to the lapse from the release of a key to the release of the other one.

The aforementioned measures can be arranged as di-graph, tri-graph, and *N*-graph latencies [18] according to the distance between the two considered letters in the typed string, as graphically outlined in Figure 6. Specifically, we refer to measures calculated from time events of two different but adjacent characters as di-graph latencies. Time features are called tri-graph when computed from time events of two characters separated by an intermediate one. Similarly, if the timing features are calculated between time events of characters distant *N*-1 positions in the provided text string, then the time features are known as *N*-graph features. The apexes *D*, *T*, and *N* are used in the following to specify the kind of considered latency, being it di-graph, tri-graph, or *N*-graph, respectively, as graphically shown in Figure 6.

2.1.3 | Typing speed

From the collected KD data, the user typing speed can be determined by means of several measures, such as the words per minute (*WPM*), defined in [19] as the average number of words that could be typed by the considered user in a minute. This value is estimated in [19] by considering the length in characters of the input string (*IL*, excluding backspaces),

TABLE 1 Summary of data available in mobile devices for keystroke dynamics-based biometric recognition systems

Type	Acronymn	Description
Timing information	*KPT*/*KRT*	Key-press/key-release timestamps
	KHT (*KDT*)	Key-hold (key-dwell) time, lapse passing from the press and release of a key
	PRT	Key press-release time, lapse passing from the press to the release of different keys
	PPT	Key press-press time, lapse passing between the press events of different keys
	RPT (*KFT*)	Key release-press (key-flight) time, lapse passing from the release to the press of different keys
	RRT	Key release-release time, lapse passing between the release events of different keys
	WPM	Word per minute, a measure of typing speed
	CPS	Characters per second, typically excluding backspaces
	AdjWPM	Adjusted word per minute, modified version of *WPM* considering errors
	KPS	Keystrokes per second, similar to CPS (including backspace)
Screen motion event (SME)	*P*	Finger touch pressure on the mobile device screen
	A	Finger area on the mobile device screen
	TC	Finger touch coordinates (x, y) on the mobile device screen
	DfC	Cartesian distance from the key centre of the touch at release event
	DbT	Cartesian distance between the coordinates of two different finger touches
	V	Velocity of the motion event, specifying the direction of finger movement on the mobile device screen
	DRG	Drag, Cartesian distance between the coordinates at press and release of a key
Sensor information motion	*ASD*	Accelerometer providing acceleration along x, y, z axes (considering gravity)
	UASD	Uncalibrated accelerometer providing acceleration along x, y, y axes (without bias compensation)
	LASD	Accelerometer providing acceleration along x, y, z axes (excluding gravity)
	GRSD	Gravitational sensor providing the magnitude and direction of gravity over x, y, z
	GSD	Gyroscope providing the rotation around x, y, z
	UGSD	Uncalibrated gyroscope providing the rotation around x, y, z axes (excluding gyro-drift compensation)
	RVSD	Rotation vector sensor providing the orientation of the mobile device
Position	*GRVSD*	Game rotation vector sensor, similar to *RVSD* (excluding the use of geomagnetic field sensor)
	GERVSD	Geomagnetic rotation vector sensor, similar to *RVSD* (including the use of geomagnetic field sensor)
	GFSD	Geomagnetic field sensor providing the strength of geomagnetic field along x, y, z
	UGFSD	Uncalibrated geomagnetic field sensor providing the strength of geomagnetic field along x, y, z axes (without calibration and with bias estimation)
	OSD	Orientation sensor providing the x, y, z mobile device position
	PSD	Proximity sensor providing the face distance from the mobile device screen
Addit.	*SCD*	Steps counter monitoring the pace of a user while walking
	SDD	Step detector, monitoring user's step movements while walking using the mobile device

the time required to type it (*IT*, in seconds, including backspaces), and assuming an average length of five characters for a word, thus obtaining $WPM = [60\ (IL - 1)]/(5\ IT)$.

Some other metrics have been mentioned in [19] for evaluating KD typing speed, such as:

- Characters per second (*CPS*), defined as $CPS = (IL - 1)/IT$
- Adjusted *WPM* (*AdjWPM*) [20], considering typing errors to improve the estimate of *WPM;*
- Keystrokes per second (*KPS*), similar to *CPS*, yet considering the backspace key as a keystroke, different from *CPS*

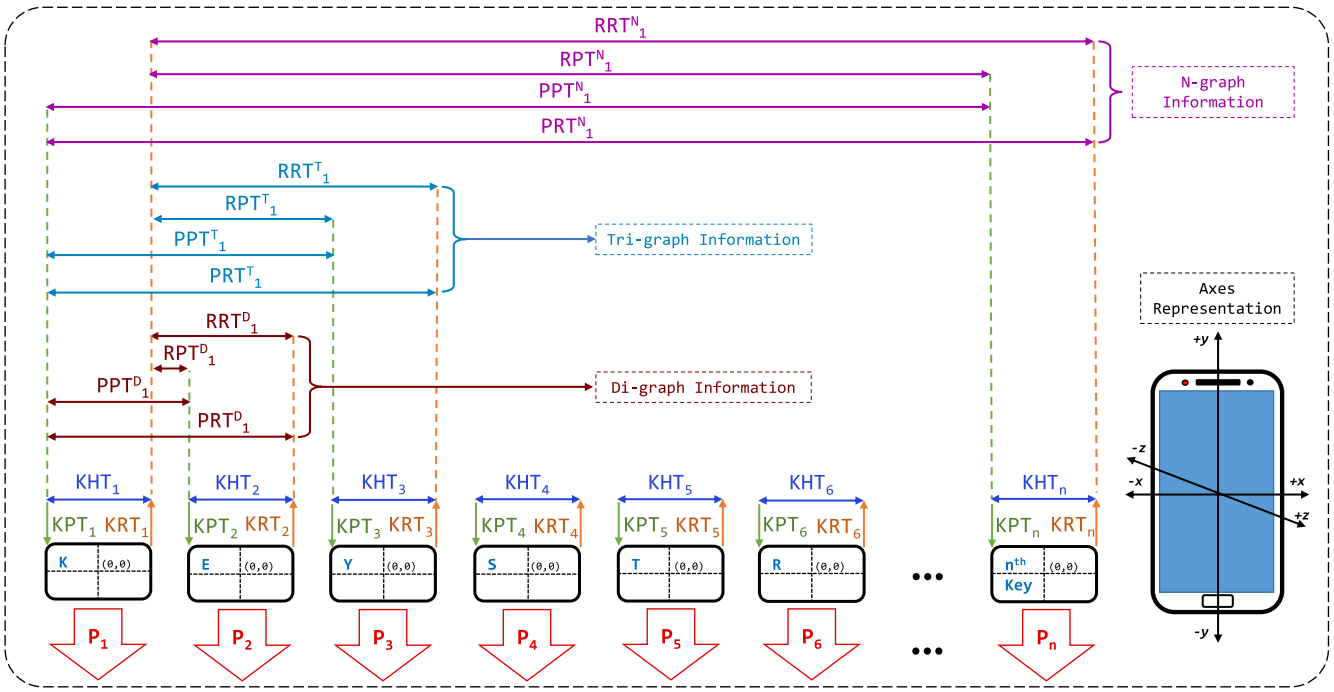

FIGURE 6 Mobile KD Signals through soft keyboard and different accessible sensors data; KD, keystroke dynamics; KHT, key-hold time; KPT, key-press time; KRT, key-release time; PPT, key press-press time; PRT, key press-release time; RPT, key release-press time

which does not consider erroneous typing and backspaces, just as *WPM*.

2.2 | Sensor information

Modern mobile devices embed several sensors which may provide data to be used for KD-based biometric recognition [21], in addition to timing information. For instance, since the emergence of smartphones with touch screens, finger pressure (*P*) and area (*A*) are two of the most-commonly exploited characteristics that are used within the KD framework [22]. Similarly, the way a user holds the phone, the amount of acceleration observed while moving, the number of fingers used during typing, are all relevant information that can be used to complement and support KD-based biometric recognition in the framework of continuous authentication [23].

The Android platform categorizes the aforementioned sensors according to three classes, that is, motion sensors, position sensors, and environment sensors[2]. Additionally, the touchscreen itself has its own set of managed events. The signals which can be recorded by a mobile device through sensors belonging to such groups, and used for biometric recognition purposes analysing the users' typing behaviour, are detailed hereafter. Environment sensors are not described here since they do not collect data regarding the users' interaction with the device.

[2]https://developer.android.com/guide/topics/sensors/sensors_overview

2.2.1 | Screen motion event sensor data

The *Screen Motion Event* (SME) class in Android OS defines a set of events which are executed when any kind of motion occurs on the touchscreen of a mobile device. Among them, *screen pressure* reports the pressure (*P*) exerted by a finger on the touch screen, with values in the range between 0 and 1. Such elements, indicated as $P_1,\ldots, P_n$ in Figure 6, actually contain two pressure values: the one applied when the finger is pressed against the touchscreen, and the one at the moment of removing the finger from the touchscreen. Pressure data have been employed for the first to design KD-based biometric recognition systems for mobile devices in [24].

When the finger touches the screen of a mobile phone, the touch *area* (*A*) can be also recorded. Two finger area values are taken for each key, at the press and release event. Generally, the value of the touch area is a float value between 0 and 1.

Similarly, also the (*x*, *y*) *touch coordinates* (*TC*) can be recorded, with the centre (0, 0) set either with reference to the whole keyboard, as typically done and shown in Figure 6, or to the centre of the pressed key, as in [25] where the distance from the centre (*DfC*) is computed from the *TC* at the release event.

The *TC* features of two different keys can be employed to compute the *distance between touches* (*DbT*) as the Cartesian distance between the coordinate positions of the considered touches.

Another motion event relates to the direction of the pointer. *Velocity* (*V*) is often an important factor in monitoring a gesture behaviour, also for checking whether a gesture occurred or not. In addition, *finger drag* (*DRG*), that is,

the measurement of the distance made by the finger from key-down event till key-up event, as well as the angle of the movement, can be exploited.

2.2.2 | Motion sensor data

The Android platform provides different types of *motion sensors*[3], which may be hardware- or software-based. For instance, three distinct types of information can be derived from the accelerometer sensor: the *standard accelerometer* (*ASD*) provides three values along (x, y, z) directions, including gravity. *Uncalibrated accelerometer* (*UASD*) provides six values, three of which measure acceleration along (x, y, z) without any bias compensation, while the other three values provide acceleration along the axes with drift compensation. The *linear accelerometer* (*LASD*) stores the acceleration along (x, y, z) direction exactly as *Accelerometer*, yet with the exclusion of gravity. The unit of these three different types of accelerometer data is m/s^2.

The *gravitational sensor* (*GRSD*) provides three-dimensional data indicating the magnitude and direction of gravitational force. This software-based sensor is generally employed to estimate the mobile device relative orientation.

The rate of rotation around a device (x, y, z) axes is measured by the *gyroscope sensor* (*GSD*). In practical scenarios, gyroscope noise and drift introduces errors, and these errors should be properly compensated. The drift and noise are estimated from information collected through several sensors, like the gravity or accelerometer ones. The *uncalibrated gyroscope* (*UGSD*) is just like the gyroscope, yet in this case the rate of rotation remains the same, with no addition of gyro-drift compensation. Still rate of rotation is compensated by adding calibration of factory level and compensation of temperature. The estimated drift around each axis could also be calculated using the uncalibrated gyroscope.

Eventually, the *rotation vector sensor* (*RVSD*) provides the device orientation, expressed through an angle θ and the (x, y, z) axes, with respect to the earth's coordinate system. This type of sensor is a virtual or software sensor which combines the values from hardware sensors, such as the accelerometer, the geomagnetic field sensor, and the gyroscope (if available), to provide an efficient and accurate orientation of the device.

2.2.3 | Position sensor data

As for the motion sensors, also the Android *position sensors*[4] may be hardware- or software-based. The *game rotation vector sensor* (*GRVSD*) can be compared to the *RVSD*, with the difference that it does not use the geomagnetic field sensor data. The y axis in this sensor does not point north, but to some other reference direction. The order of drift magnitude of the gyroscope around the z axis is the same as the drift of the reference direction. Since this sensor does not use the magnetic field, relative rotations are more accurate in comparison to other rotation sensors.

The *geomagnetic rotation vector sensor* (*GERVSD*) is also similar to the *RVSD*, yet in this case the sensor uses a geomagnetic field sensor instead of a gyroscope. Less power is thus consumed, but while its advantage is low power consumption, there is a disadvantage in terms of less accuracy. The geomagnetic field sensor can also provide two different measures of the geomagnetic field: the *geomagnetic field sensor* (*GFSD*) gives geomagnetic field strength in three axes, while the *Uncalibrated Geomagnetic Field Sensor* (*UGFSD*) does not include hard iron calibrations.

The *orientation sensor* (*OSD*) is a virtual sensor estimating mobile device location keeping Earth magnetic north pole as reference. It usually calculates the angles of orientation by combining the values from the geomagnetic field sensor and the accelerometer sensor. Using the data from these sensors, the following information can be provided:

- *Azimuth* (rotation degrees about the z axis), angle between the north and the device compass direction in the current state;
- *Pitch* (rotation degrees about the x axis), angle between planes parallel to the ground and to the device screen;
- *Roll* (rotation degrees around the y axis), angle between planes perpendicular to the ground and to the device screen;

It is worth mentioning that the combination of accelerometer and geomagnetic field data requires high computational power, with the consequence that the achievable orientation sensor precision and accuracy is typically low. Specifically, this measure can be trusted only when the roll angle is 0, otherwise its use is not recommended.

The *proximity sensor* (*PSD*) monitors the distance between the face and the mobile phone, with a maximum range of about 50 cm. It can produce signals comprising either float values indicating the estimated distances, or binary terms describing whether the user is near or far from the screen.

2.2.4 | Additional sensor data

There is additional sensor information, typically classified within the motion sensors class, not yet exploited in literature about KD for biometric recognition, which could be useful to extract discriminative information from a typing behaviour. For instance, the *step counter* (*SCD*) and *step detector* (*SDD*) sensors, both hardware-based sensors, provide information regarding human steps data. These could be useful to monitor KD while walking. The *SCD* provides the number of steps the user of the device has taken since it has been switched on, and the *SDD* triggers an event when a user starts a step.

The timing information mentioned in Section 2.1, and the sensor information reported in Section 2.2, can be exploited by arranging them in different array-wise data types collected for

[3]https://developer.android.com/guide/topics/sensors/sensors_motion
[4]https://developer.android.com/guide/topics/sensors/sensors_position

different pressed keys, with this collection of arrays employed as features vectors, as discussed in Section 4.

3 | MOBILE BASED KEYSTROKE DATASETS

A general overview of the databases collected and exploited in state-of-the-art KD-based biometric recognition systems for mobile devices is presented in this section. Table 2 provides a summary of the main datasets collected in literature, together with their primary characteristics. In more detail, only datasets comprising KD samples acquired from at least 20 subjects are there reported. The following subsections detail the properties of each collection of data, categorizing them according to the kind of input recorded from the involved subjects.

3.1 | Fixed text

As mentioned in Section 2, fixed text involves the use of PINs, passwords, or passphrases, detailed in the following.

3.1.1 | Personal identification numbers

PINs have been always used in the access control protocols of mobile devices, mainly to unlock SIMs. They are also often required to access online services such as bank accounts. Not by chance, most of the research on KD-based biometric recognition for mobile devices has focused on PINs as input. The first proposal of a recognition system for mobile devices relying on KD has actually investigated the discriminative capability of the behaviour recorded while typing four- and 10-digit PINs [15], collecting data from 16 subjects using a modified handset connected with a desktop computer.

The first dataset comprising PINs collected through a mobile phone has been presented in [26]. Different phones have been yet used by each of the 25 involved subjects, with this aspect possibly affecting the computed recognition performance. All the employed devices were equipped with a hard keypad, with keys arranged in a 4 × 3 matrix. The same type of keyboard was embedded in the Samsung SCH-V740 model, employed in [27] to record KD from 25 subjects, each donating 30 KD samples. In order to increase typing uniqueness, temporal cues, working like a metronome, have been provided during acquisitions to help subjects keeping the beat. Given the limited capabilities of the employed devices, only timing information has been collected in both aforementioned studies.

Since then, research on mobile KD biometrics has been performed on data collected using smartphones with soft keyboards. However, it has to be mentioned that, when using the Android platform and factory designed soft keyboards such as *Gboard*, it is not possible to extract the needed timing information since the function is inhibited for privacy reasons. Therefore, researchers are required to develop custom soft keyboards using the available support[5], install them on the mobile device, and thus record and get access to the desired KD data. This is typically done trying to resemble the layout of the default soft keyboards available in smartphones [59].

Mobile devices with touchscreens have been employed to collect KD data in [28], thus allowing to record SMEs related to the applied finger pressure and finger area. Recordings from multiple sessions have been there considered for the first time, with two PINs acquired from each of 100 subjects during each week of the experiment, which has lasted for 5 weeks. The employed inputs had varying lengths, ranging from four to eight digits, with most of subjects (44) using 4-digit PINs. Accelerometer data have been collected for the first time in [29], where also timing and finger area information have been considered. Data from 55 subjects, each contributing with 30 KD samples, have been there recorded. This database has been also employed for the tests described in [30].

Five different PINs, three of them with four digits and the other ones with eight, have been collected in [31] from 80 subjects, each contributing with 25 samples. Both normal and extreme cases have been considered, with PINs 3-2-4-4, 1-2-5-9-7-3-8-4, and 1-2-5-9-8-4-1-6 representing the former scenario, while 1-1-1-1 and 5-5-5-5 have been used for the latter one. The recorded signals comprise timestamps, finger pressure and area, as well as accelerometer data.

The same set of information has been also collected in [32], where 25 PINs with four digits have been recorded from each of 80 subjects.

The first public database[6] containing KD recorded while using PINs has been described in [33]. A Samsung Galaxy Tab with a 10.1-inch screen has been there employed to collect four- and 16-digit PINs from 150 users, each providing 10 inputs for both short (‘5560’) and long PINs (‘1379666624680852’). Timing, finger pressure, and finger area information have been acquired, yet only timing and pressure data have been employed in [33], while finger area has been exploited only in a successive work from the same authors [35]. Data collected in [33] have been also employed in [34, 35].

Another public database[7] has been collected using an Android LG-D820 Nexus 5 mobile device in [37], where a 10-digit PIN (‘9141937761’) has been acquired for 30 times from each of 52 subjects. In addition to timing, finger pressure and area, and accelerometer data, for the first time in studies dealing with PINs also *TC*, and data from the gyroscope sensor, have been also recorded.

An even larger list of sensors has been considered in [38], where a 6-digit PIN (‘766420’) has been collected for 100 times from each of 20 subjects using a Nexus 5X phone. Specifically, the acquired data comprise timing information, finger pressure and area, TC, accelerometer and linear accelerometer, rotation and game-rotation, gyroscope, and uncalibrated gyroscope.

[5] https://android.googlesource.com/platform/packages/inputmethods/LatinIME/
[6] https://sites.google.com/site/touchstrokedynamics/
[7] https://bitbucket.org/pacebiometrics/android-biokeyboard/src/master/

TABLE 2 Databases collected for evaluating KD-based biometric recognition on mobile devices

Type		Paper	Subjects	Postures	Sessions[a]	Samples[b]	Input	Signals	Availability
Fixed text	PIN	Zahid et al. 2009 [26]	25	1	1	n/a	8-digit	Timing	Private
		Hwang et al. 2009 [27]	25	1	1	35	4-digit	Timing	Private
		Tasia et al. 2013 [28]	100	1	5	2	4- to 8-digit	Timing, SME	Private
		Ho et al. 2013 [29]	55	1	1	30	4-digit	Timing, SME, motion	Private (also used in [30])
		Zheng et al. 2013 [31]	80	1	1	25	4- and 8-digit	Timing, SME, motion	Private
		Mendizabal et al. 2014 [32]	80	1	1	25	4-digit	Timing, SME, motion	Private
		Teh et al. 2015 [33]	150	1	1	20	4- and 16-digit	Timing, SME	Public (also used in [34–36])
		Coakley et al. 2016 [37]	52	1	1	30	10-digit	Timing, SME, motion	Public (also used in [36])
		Lee et al. 2019 [38]	20	1	1	100	6-digit	Timing, SME, motion, Position	Private
		Wang et al. 2019 [39]	104	1	1	20	4-digit	Timing, SME, motion	Private
	Password	Campisi et al. 2009 [40]	30	1	1	120	6-character	Timing	Private
		Antal et al. 2014 [41]	42	1	2	20	14-character	Timing, SME	Public (also used in [42])
		El-Abed et al. 2014 [43]	53	1	3	5	14-character	Timing	Public (also used in [42])
		Giuffrida et al. 2014 [44]	20	1	1	40	8-character	Timing, SME, motion	Private
		Buschek et al. 2015 [25]	28	3	2	120	10-character	Timing, SME	Private
		Al-Obaidi et al. 2016 [45]	56	1	2	17	10-character	Timing, SME	Private
		Antal et al. 2016 [46]	54	1	3	60	13- to 15-character	Timing, SME, motion	Public (also used in [36])
		El-Abed et al. 2018 [47]	47	4	1	15	14-character	Timing	Public
		Tse et al. 2020 [48]	31	1	1	50	14-character	Timing, SME	Private
	Passphrase	Trojahn et al. 2013 [49]	152	1	1	10	17-character	Timing	Private
		Gascon et al. 2014 [50]	315	1	1	1/10	160-character	Motion, Position	Private
		Kambourakis et al. 2014 [51]	20	1	1	12	47-character	Timing, SME	Private
Free Text		Clarke et al. 2003 [15]	32	1	3	10	14-word (average)	Timing	Private
		Feng et al. 2013 [52]	40	1	1	varying	23 words (average)	Timing, SME	Private
		Sun et al. 2017 [53]	26	1	varying	>29	varying	Timing, SME, motion	Private
		Crawford et al. 2017 [11]	36	6	1	22	varying	Timing, motion, Position	Private
		Inguanez et al. 2017 [54]	32	1	1	75	2- to 13- character	Timing, SME	Private
		Alshanketi et al. 2019 [55]	100	1	2	10	6-digit	Timing, SME	Private

TABLE 2 (Continued)

Type	Paper	Subjects	Postures	Sessions[a]	Samples[b]	Input	Signals	Availability
	Cilia et al. 2018 [56]	24	2	1	1/23	15-sentence	Timing, SME	Private
	Buschek et al. 2018 [57]	30	6	varying	varying	varying	Timing, SME	Public
	Belman et al. 2019 [58]	117	1	2	10	>50-character	Timing, SME, motion	Public

Abbreviations: KD, keystroke dynamics; SME, screen motion event.

[a]per posture.

[b]per posture and session.

PIN acquisitions have been taken for the first time according to multiple postures in [39], where lying, sitting, standing, and walking conditions have been considered. At least 20 KD samples have been collected from each of 104 subjects, considering timing, finger pressure and area, and accelerometer signals.

3.1.2 | Password

A Nokia 6680 equipped with a hard keypad has been employed in the first detailed evaluation of the discriminative capability of KD recorded while typing passwords [40]. Specifically, six different passwords, each made of ten characters, have been collected from 30 subjects, providing 20 samples for each password. The employed keypad had characters multiplexed in a 4 × 3 matrix, with more characters associated to the same key. As for the other studies conducted on mobile devices with hard keypads, only timing information has been acquired.

A public database[8] has been described in [41]. Two acquisition sessions have been carried out, with 42 subjects asked to provide 30 passwords in each session. Five subjects have used a Mobil LG Optimus L7 II P710 phone, while 37 other subjects have used a Nexus 7 tablet. The password '.tie5Roanl', requiring 14 key presses due to the need for switching two times between upper- and lower-case letters, and two times between letters and numbers, has been employed. The collected information consists of timing data, together with finger pressure and area, and have been used also in the work [60].

Another public dataset[9] has been presented in [43], where 53 individuals have participated in the recording process, typing the password 'rhu.university' for five times during each of three acquisition sessions, with time periods between 3 and 30 days separating each session. Only timing information has been there collected and made available. Yet, no test has been performed on the collected data.

Forty subjects, each typing 20 passwords, have participated in the collection of the dataset described in [44]. Gyroscope information has been there considered for the first time in the KD-based biometric recognition framework, together with timing and accelerometer information.

A detailed study on KD data recorded having subjects using only a thumb, two thumbs, or the index finger for typing, has been conducted in [25]. A Nexus 5 phone held in portrait orientation has been used to record 10-character passwords from 28 subjects during two sessions, taken one week apart. For each hand posture, participants have typed six different passwords in random order, 20 times each. In addition to timing and pressure data, also touch locations have been recorded during each acquisition. The collected database has been used to perform the first proper evaluation of performance achievable using KD on mobile devices in within- and across-sessions comparison conditions.

A public dataset[10] has been also outlined in [45], where 10-character passwords have been collected from 56 subjects. Specifically, two distinct recording sessions have been there carried out, the first one using a Nexus 7 tablet, and the second one with a Nexus 9 model. The two sessions have been respectively employed for enrolment and verification purposes in the performed tests, therefore implementing an across-session and across-device experimental framework using the recorded timing, finger pressure, and finger area information.

The authors of [41] have described another public database[11] in [46]. Three different types of passwords, namely an easy (E: 'kicsikutyatarka'), a strong (S: '.tie5Roanl'), and a logically strong (LC: 'Kktsf2!2014') ones, have been collected from 54 subjects, each typing for 20 times each password during three distinct acquisition sessions, using a Nexus 7 tablet. With respect to [41], accelerometer and TC have been also recorded. The data in [46] has been also employed in [36].

Four different conditions have been instead considered in the public database[12] described in [47], where 47 individuals have participated in an acquisition campaign by typing the same password used in [43] on both a phone (Nexus 5) and a tablet (Samsung Galaxy Note 10.1), in both portrait and landscape orientations. Only timing information has been there recorded. As for [43], no test has been performed on the collected data.

The same password employed in [41] has been also used in [48], with 31 subjects each providing 50 samples, from which timing and touch coordinate information have been recorded.

[8]https://ms.sapientia.ro/~manyi/keystroke.html

[9]www.coolestech.com/RHU-Keystroke

[10]https://sites.google.com/site/keystrokedatasets/

[11]https://ms.sapientia.ro/~manyi/mobikey.html

[12]www.coolestech.com/keystroke-web-visualizer

3.1.3 | Passphrase

Long fixed texts have been considered as input of the proposed KD-based biometric recognition systems for mobile devices only in few works. Actually, such modality is probably the least suitable to be considered for practical applications.

A 17-character passphrase has been recorded from 152 subjects, each typing the same sentence for 10 times using a Galaxy Nexus, in [49], where only timing information has been collected.

A predefined text made of 160 characters, comprising different pangrams containing all letters of the English, has been considered in [50]. Recordings from 315 subjects have been taken, yet only 12 of them, employed as authorized users in the performed tests, have provided data for 10 times, while the remaining participants typed the employed text only once. Instead of resorting to standard signals such as timing, the tests carried out in [50] have exploited accelerometer, gyroscope, and orientation sensors.

A 47-character phrase, including all 26 possible letters of the alphabet ('the quick brown fox jumped over the lazy ghost'), has been instead used in [51]. Twenty subjects have typed the required inputs for 12 times on a Sony Ericsson Xperia mobile phone, while recording timing and TC information.

3.2 | Free text

Performing automatic recognition on the basis of the KD recorded while typing free text is commonly considered a much more difficult task, with respect to the exploitation of fixed text. Nonetheless, being able to recognize users in dynamic text conditions would allow performing continuous authentication, with legitimate subjects recognized while using mobile devices for tasks such as writing email or text messages, activities which are nowadays daily performed by every smartphone user. The research on this topic is therefore extremely relevant, with several approaches proposed especially in recent years.

It has however to be remarked that free text recognition has been considered in one of the very first attempts of performing KD-based recognition on mobile devices. The authors of [61] have in fact collected texts composed by a mixture of quotes, lines from movies, and typical messages, from 32 subjects, each typing 10 messages during three acquisition sessions. The typed sequences had varying length, with an average of 14 words per text. Data have been acquired through the same modified handset employed in [15], being able to acquire only timing information.

Also pressure data have been instead collected in [52], where post-login attacks have been considered. Sentences having a length varying from 14 to 53 words, and an average of 23 words, have been shown to the 40 involved subjects, then asked to enter the required sentences using the virtual keyboard.

An evaluation study lasting 8 weeks and involving 26 subjects has been described in [53]. Participants have been asked to normally use a mobile phone during the experiment, with the most active one typing 4702 messages, and the least active using the phone for 29 times. The recorded instances include both simple messages as well as inputs requiring alphabets and special characters. For each text, information regarding the timing and TC of pressed keys are collected together with the accelerometer information.

Texts have been recorded from subjects holding devices in either portrait or landscape orientations, and typing while either seated, standing, or walking, in [11]. Each of 36 participants has provided at least 22 text messages in each of six considered experimental conditions. As in [52], the involved subjects have been asked to replicate sentences shown in a box, thus performing a transcription task, which may have an impact on their typing patterns. Timing, device orientation, user position, and instantaneous gyroscope data have been recorded.

The study on [54] has been carried out on data gathered using a Samsung Galaxy S5 device from 32 users, asked to input 60 different words, with five words for each of different lengths, ranging from two to 13 characters. Furthermore, a set of 15 questions requiring participants replying with words from memory, rather than copy them from screen, has been submitted to the involved subjects. Timing, finger area, and TC have been recorded during typing activity.

OTPs have been instead considered for the first time in [55]. This kind of entry can be considered as a special case of free text, where six-digit randomly generated PINs are used to perform recognition only once, being therefore impossible to train a given classifier over data comprising the same characters. One hundred subjects have been involved for data collection, with each of them providing at least 10 OTPs during two acquisition sessions, while timing information has been collected together with finger area and pressure.

Typing sessions made up of 15 sentences, each having around six terms without digits, have been performed in [56] using a Samsung Galaxy S5 phone by each of 24 subjects. Only one of them, employed as legitimate user in the performed experimental tests, participated in multiple sessions, while the remaining ones performed only a single recording session. Two typing conditions, involving the usage of one hand or two hands for typing, have been considered, collecting timing and TC information during each session.

As in [53], free typing in the wild has been performed by 30 subjects during 3 weeks in [57], where six different postures, involving for instance typing with either two thumbs or only one, have been considered. The collected data comprise timing and touch coordinate information and are available upon request[13].

Finally, a public database[14] has been recently proposed in a pre-print [58]. The described database comprises free text samples acquired from 117 subjects using a HTC-Nexus-9 tablet and both a Samsung-S6 phone and an HTC-One mobile phones. Specifically, two sets with ten questions each, requiring

[13] http://www.medien.ifi.lmu.de/research-keyboard//intro
[14] http://dx.doi.org/10.21227/rpaz-0h66

varying cognitive loads, have been presented to the participants, which have been asked to reply with a minimum of 50 characters. A first set has been presented on a tablet, while one of the two mobile phones has been used when replying to the second set. The performed KD have been recorded through timing, finger area and pressure, touch location, accelerometer, and gyroscope information. As for [43, 47], no test has been performed on the collected data.

4 | TEMPLATE GENERATION

In this section, with reference to Section 2 and to Figure 6, the features extracted from the collected signals, and used to represent KD information through discriminative templates, are discusses. In more detail, the approaches employed to deal with fixed text inputs are outlined in Section 4.1, while the features exploited to represent free text are described in Section 4.2. A summary of the features employed in state-of-the-art studies is reported in Table 3.

4.1 | Features employed for fixed text representation

Treating KD associated to fixed text entries has the notable advantage of having to process, for each acquisition, the same and known number of input keys. Generating ordered representations of the recorded data is then generally quite straightforward, since a template can be simply generated by collecting several measures from each key-related event.

The most common data correspond to timing information, which in fact has been exploited in all the proposed works, with the only exception of [50]. In more detail, the *KHT* and di-graph RPT^D, respectively representing the hold time for each pressed key, and the flight time between consecutive keys, are considered among the most discriminative features which can be derived from timing information, and have been therefore always included in the feature sets employed in literature. As for RPT^D, it has to be mentioned that a customized version of this feature has been employed in [26], where a 4 × 3 hard keypad has been employed as user interface, and di-graphs have been categorized into measures computed between horizontal/vertical and adjacent/non-adjacent keys, instead of simply between consecutive keys as in all the rest of literature.

Among the remaining di-graph features, PPT^D is the one employed in most of the proposed works, followed by RRT^D, with PRT^D instead exploited only in four studies, as indicated in Table 3. Comparative evaluations have however pointed out that all di-graph features contain similar amounts of discriminative information [33]. It has been instead often noticed that the hold time *KHT* has less discriminative capability than di-graph latency times [33, 36], although there is not a general consensus [44]. Several concurrent evidences have been reported regarding the low discriminative power of tri-graph and *N*-graph latencies [35, 36, 44, 49]. Therefore, literature studies suggest to take into account only hold time and di-graph latencies when extracting features from the collected timing information.

Besides temporal data, finger pressure and area recorded by touchscreen devices have been exploited in most cases. Both features, but especially pressure, have shown discriminative capabilities better than timing characteristics in the vast majority of the works where they have been used [25, 29, 33, 35, 36], although results conflicting with this observation have been also reported [37].

SME signals collected by touchscreen devices also include *TC*, recorded at both key press and key release instants. The collected information has been used as-is [25, 37, 38, 48], or exploited to compute additional metrics, such as the drag *DRG* between TC at press and release events of the same key [25], the distance *DbT* between coordinates of consecutive key press events [25, 46, 51], and also the velocity *V*, computed as quotient of the distance *DbT* and the corresponding latency time [46, 51].

As previously mentioned, the aforementioned timing and SME data are collected at specific key-related events and then arranged into a template whose size depends on the length of the processed input. It is yet also possible to generate representations not depending on the length of the recorded KD. Such second-order features [41, 46] are typically obtained computing statistics from the collected raw data, including for instance the minimum, the average, the maximum, and the standard deviation metrics of the considered features [32, 35, 45, 48]. Exploiting such second-order features has always resorted in improved recognition performance, being also possible to use fixed-size templates relying only on them [46].

Statistical measures have been often employed to process signals acquired through inertial sensors, such as accelerometer, gyroscope, and rotation. These data are in fact continuously recorded by mobile devices, and therefore not necessarily associated to key press and release events. In order to derive fixed-size representations from the collected data, statistical features have been actually computed in [29, 44], and [30]. Given the long length of the passphrases considered in [50], the nine signals collected from the 3-axis accelerometer, gyroscope, and rotation sensors are divided into frames lasting a fixed amount of time, and features are then computed from them. Differently from the aforementioned approaches, and similarly to the methods employed to process timing and SME information collected from fixed-text inputs, several studies exploit the signals recorded from motion and position sensors by sampling them at key press and release events, as done in [32, 37, 39, 46], and [38].

4.2 | Features employed for free text representation

While all approaches dealing with KD derived from fixed text employ similar methods for generating the used biometric representations, either collecting signals at specific and expected time events, or computing statistical features summarizing the observed behaviour, more heterogeneous

TABLE 3 Used features, considered modalities, and attained performance in state-of-the-art methods performing KD-based biometric recognition on mobile devices

Type		Modality	Study	Features	Classifier	Template Size	INPUT	Performance			
								FAR	FRR	EER	CIR
Fixed text	PIN	Identification	Mendizabal et al. 2014 [32]	*KHT*, RPT^D, *P*, *A*, *LASD*	PCA and MLP	63	4-digit	-	-	-	90.0%
		Closed-set verification	Zahid et al. 2009 [26]	*KHT*, RPT^D	PSO and GA	13	8-digit	2.1%	0.0%	-	-
			Ho et al. 2013 [29]	*KHT*, RPT^D, *A*, *ASD*	SVM	35	4-digit	4.4%	5.3%	-	-
			Deng et al. 2015 [30]	*KHT*, RPT^D, *A*, *ASD*	DNN with RBM	35	4-digit	-	-	2.8%	-
			Coakley et al. 2016 [37]	*KHT*, RPT^D, PPT^D, *P*, *A*, *TC*, *DRG*, *DbT*, *GSD*	SVM	614	10-digit	-	-	3.9%	-
			Wang et al. 2019 [39]	*KHT*, RPT^D, *P*, *A*, *ASD*	SVM	146	4-digit	-	-	0.1%–2.1%	-
		Open-set verification	Hwang et al. 2009 [27]	*KHT*, RPT^D	Distance	7	4-digit	-	-	4.0%	-
			Tasia et al. 2013 [28]	RPT^D, PPT^D, *P*, *A*	Distance	18 to 38	4- to 8-digit	-	-	8.4%	-
			Zheng et al. 2014 [31]	*KHT*, RPT^D, *P*, *A*, *ASD*	Distance	63	4-digit	-	-	7.3%	-
						127	8-digit	-	-	3.6%	-
			Teh et al. 2015 [33]	*KHT*, RPT^D, PPT^D, RRT^D, PRT^D, *P*	Gaussian	20	4-digit	-	-	8.5%	-
						92	16-digit	-	-	5.5%	-
			Maiorana et al. 2019 [34]	*KHT*, RPT^D, PPT^D, RRT^D, PRT^D, *P*	CNN	24	4-digit	-	-	4.5%	-
						120	16-digit	-	-	2.8%	-
			Lee et al. 2019 [38]	*KHT*, RPT^D, PPT^D, RRT^D, PRT^D, *A*, *TC*, *ASD*, *LASD*, *UGSD*, *GSD*, *RVSD*, *GRVSD*	Distance	602	6-digit	1.0%	11.1%	-	-
			Teh et al. 2019 [35]	*KHT*, RPT^D, RPT^I, RPT^N, PPT^D, RRT^D, PRT^D, *P*	OC-SVM	115	4-digit	-	-	9.9%	-
						187	16-digit	-	-	7.1%	-
	Password	Identification	Antal et al. 2014 [41]	*KHT*, RPT^D, PP^D, *P*, *A*	RF	71	14-character	-	-	-	93.0%
			Tse et al. 2020 [48]	*KHT*, RPT^D, PP^D, RR^D, *TC*	LDA	116	14-character	-	-	-	88.3%
		Closed-set verification	Alshanketi et al. 2016 [42]	*KHT*, RPT^D, PPT^D, RRT^D, *P*, *A*	RF	85	14-character	-	-	5.8%	-
		Open-set verification	Campisi et al. 2009 [40]	*KHT*, RPT^D, PPT^D, RRT^D	Distance	37	10-character	-	-	13.1%	-
			Giuffrida et al. 2014 [44]	*KHT*, RPT^D, RPT^I, RPT^N, *ASD*, *GSD*	OC-SVM	n/a	8-character	-	-	0.1%	-
			Buschek et al. 2015 [25]	*KHT*, RPT^D, PPT^D, RRT^D, *TC*, *DRG*, *DbT*, *P*	Gaussian, LSAD	240	10-character	-	-	3.8%	-
			Al-Obaidi et al. 2016 [45]	*KHT*, RPT^D, PPT^D, *P*, *A*	Distance	71	10-character	-	-	4.9%	-

TABLE 3 (Continued)

Type	Modality	Study	Features	Classifier	Template Size	INPUT	Performance			
							FAR	FRR	EER	CIR
		Antal et al. 2016 [46]	*KHT*, RPT^D, PPT^D, *P*, *A*, *TC*, *DbT*, *V*, *ASD*	Distance	72	13-character (S)	-	-	14.3%	-
					72	14-character (LS)	-	-	12.9%	-
					82	15-character (E)	-	-	16.0%	-
		Kalita et al. 2020 [36]	*KHT*, RPT^D, PRT^D	GMM	82	14-character (LS)	-	-	2.3%	-
	Passphrase	Gascon et al. 2014 [50]	*ASD*, *GSD*, *OSD*	SVM	2376	160-character	1.0%	8.0%	-	-
		Open-set verification	Trojahn et al. 2013 [49]	*KHT*, RPT^D, RPT^T	Distance	48	17-character	4.2%	4.6%	-	-
			Kambourakis et al. 2014 [51]	*KHT*, RPT^D, *DbT*, *V*	*k*-NN	185	47-character	23.7%	3.5%	-	-
Free text		Identification	Sun et al. 2017 [53]	*KHT*, RPT^D, *DfC*, *ASD*	GRU-BRNN	varying	varying	-	-	-	82.7%
		Closed-set verification	Clarke et al. 2007 [61]	*KHT*	RBF	2-6	14-word (avg.)	-	-	18.0%	-
			Feng et al. 2013 [52]	*KHT*, RPT^D, *P*	RF	122	23-word (avg.)	1.0%	1.0%	-	-
			Crawford et al. 2017 [11]	*KHT*, RPT^D, *GSD*, *OSD*	DT and LR	¿4	varying	1.8%	5.5%	-	-
			Inguanez et al. 2017 [54]	*KHT*, RPT^D, *TC*, *A*, *DRG*, *V*	MLP	47	2- to 13-character	6.3%	4.9%	-	-
			Alshanketi et al. 2019 [55]	*KHT*, RPT^D, RRT^D, *P*, *A*	RF	n/a	6-digit	-	-	22.9	-
			Cilia et al. 2018 [56]	*KHT*, RPT^D, *TC*, *DRG*, *V*, *DbT*	SVM	n/a	15-sentence	-	-	0.4%	-
		Open-set verification	Buschek et al. 2018 [57]	*KHT*, RPT^D, *TC*, *DfT*, *DRG*	Gaussian	8	varying	-	-	14.7%	-

Abbreviations: A, area; ASD, standard accelerometer; CNN, convolutional neural networks; DbT, distance between touches; DRG, finger drag; DNN, deep neural network; GA, genetic algorithm; GMM, Gaussian models; GRU-BRNN, gated recurrent unit bidirectional recurrent neural network; GRVSD, game rotation vector sensor; GSD, gyroscope sensor; KHT, key-hold time; *k*-NN, *k*-nearest neighbour; KPT, key-press time; KRT, key-release time; LSAD, least squares anomaly detector; LDA, linear discriminant analysis; MLP, multi-layer perceptron; P, pressure; PCA, principal component analysis; PPT, press-press time; PRT, press-release time; PSO, particle swarm optimization; RBF, radial basis function; RBM, Gaussian restricted Boltzmann machine; RF, random forest; RPT, release-press time; RRT, release-release time; RVSD, rotation vector sensor device; SVM, support vector machine; TC, touch coordinates; UGSD, uncalibrated gyroscope; WPM, word per minute.

approaches have been proposed to process free text inputs. In fact, in this latter case the signals collected during enrolment and recognition stages can be considerable different, in term of content and length. In principle, it is not even guaranteed that a subject types the same keys during the two phases. Therefore, specific strategies have to be defined to generate templates which could be compared to determine their similarity.

Some approaches have been defined to select, from the collected data, a subset of events which are most likely to occur several times while typing free texts. For instance, only the most recurrent characters have been employed in [61] to perform recognition regardless the actual words or sentence composed by users. For such characters, which have been identified as the set {E,T,A,O,N,I}, the hold time *KHT* has been computed and used as discriminative feature. An analogous approach has been followed in [52], where the 40 most frequently used combinations of consecutive keys have been chosen according to the statistical distributions of English words. The RP^D di-graph latencies associated to such pairs have been used to create the desired templates, together with *KHT* and pressure information associated to each of the 41 keys available in the employed keyboard. A dynamic feature space has been considered in [11], where hold times, di-graph latencies, and instantaneous gyroscope, orientation, and position, for which at least a minimum number of four different instances are available, have been used at the early stage of the data collection process, with the feature space growing as more data is being collected. Only the first two pressed keys of each typed word have been instead considered in [54] to define the employed template, together with statistical metrics of all the collected di-graphs, coordinates, distance, velocity, and finger area.

Conversely, the entire amount of collected data is employed in [53], without performing any selection, by directly storing the time series associated to the duration of a keystroke, the time since last keystroke, and the distance from last keystroke. The recorded sequences are divided into different 'views', depending on whether the performed keystroke is associated to an alphabet letter or to a special character. A third view consisting of the time series recorded by the accelerometer sensor has been then added. Statistical features associated to all pressed keys and related to the corresponding *TC*, drag distance and velocity, and *KHT*, have been instead considered in [56]. Distributions of data have been also estimated as in [55, 57]. In the former case, the layout of the employed virtual keyboard has been considered to categorize consecutive keys into pairs having similar distances in either left or right directions. The flight times corresponding to each occurrence of the so-defined pairs are then averaged to assign representative features to each data point, thus creating graphical representations of the collected data having travel distances on the x-axis, and corresponding flight times on the y-axis. Missing points which could not be evaluated on the basis of the collected data are instead estimated resorting to a polynomial curve fitting algorithm. As for [57], data recorded during a typing activity are represented through distributions of *KHT*, di-graph latencies, *TC*, and drag characteristics. Gaussian distributions are used to model the distribution of the collected samples. In order to exclude breaks in the typing process, a maximum typing gap time of four seconds is set, after which the related keystrokes are not considered for computing the considered statistics.

5 | TEMPLATE COMPARISON AND DECISION MAKING

The template comparison and decision-making strategies used in state-of-the-art KD-based recognition methods are here described. Section 5.1 outlines the approaches employed to compare templates derived from fixed texts, while the strategies adopted for dealing with free texts are presented in Section 5.2. In both cases, the proposed methods have been categorized into identification, closed-set verification, and open-set verification modalities. A summary of the employed methods is reported in Table 3, where the approaches resulting in the best achieved performance for each work are mentioned.

Identification scenarios are adopted in systems requiring data from all legitimate users being available for training the employed classifiers. The recognition phase is then carried out selecting probe samples from the considered users, and taking a decision about the identity of their owners exploiting the trained machine learning approaches, therefore performing 1-to-many comparisons.

Verification consists in deciding whether a probe sample has been collected from the user whose identity has been declared, comparing the available data with a model previously stored during the enrolment phase of the considered user. A closed-set scenario requires, during the enrolment stage of each user, the availability of data taken from all the subjects that are considered for testing purposes, similarly to identification conditions. According to such approach, during enrolment it is possible to train binary classifiers, discriminating samples of the considered user from those of all the rest of exploited subjects. Such possibility is instead not allowed in open-set verification, where data taken only from the interested user is available to perform the enrolment phase, and samples from subjects not available during enrolment are then employed to evaluate the capability of rejecting unauthorized individuals in the recognition phase. One-class classifiers, often indicated also as anomaly detectors, have to be trained for each user during enrolment in this scenario, with the aim of modelling the distribution of available KD samples.

Within the context of KD-based biometric recognition on mobile devices, the scenario most resembling practical usage is represented by the open-set verification, since it is unrealistic to assume that data taken from all the subjects which will attempt accessing a mobile device or its applications are available during the enrolment of its legitimate user.

5.1 | Fixed text comparison

As reported in Table 3, among the surveyed studies, identification scenarios have been considered once for PIN inputs [32], and twice for passwords [41, 48]. Several machine learning approaches from the Waikato Environment for Knowledge Analysis (WEKA) suite of algorithms [62], namely naive Bayes, Bayeasian networks, k-nearest neighbour (k-NN), classification and regression trees (CART), logistic regression (LR), linear discriminant analysis (LDA), support vector machines (SVMs), and multi-layer perceptron (MLP), have been tested in [41, 48] to achieve the best possible recognition results, with random forest (RF) outperforming other approaches in [41], and LDA giving the best results in [48]. A MLP network with a single hidden layer has been instead fed with representations made of 20 coefficients, obtained applying principal component analysis on the employed templates, in [32].

Closed-set verification scenarios have been applied to PIN inputs in [26, 29, 30, 37], and [39], to passwords in [42], and to passphrases in [50]. Particle swarm optimization (PSO) is used in conjunction with genetic algorithms to classify the timing features employed in [26]. SVMs have been employed in [29, 37, 50], and [39]. In this latter case, an adversarial noise-based approach, with noise added to the available samples to increase the generalizability of the trained classifiers, has been also proposed. RF has been exploited in [42], where an analysis on the effectiveness of sampling the distributions of genuine and impostor classes employed during the enrolment of each user is also performed. A deep neural network with two hidden layers, initialized on the weights obtained by training Gaussian restricted Boltzmann machines in an unsupervised modality, has been employed in [30].

Most of studies performing open-set verification with fixed-text KD inputs have resorted to distance-based approaches, computing either Euclidean, Manhattan, or Mahalanobis distances between a probe sample and the enrolment set [27, 28, 31, 38, 40, 45, 46, 49]. The computed distances are then compared to a threshold to verify the identity of the subject. A user-specific normalization of the distances obtained from different features has been proposed in [40], while user- and feature-specific thresholds have been instead applied to the computed dissimilarity scores, before taking the final decision, in [38, 45, 49]. Gaussian models have been employed to represent the considered feature distributions, with similarity scores computed evaluating the estimated distributions at points given by the probe samples, in [25, 33]. In the latter case, also a least squares anomaly detector (LSAD) has been employed. A similar approach, yet resorting to mixture of Gaussian models, has been also used in [36]. One-class SVMs, which exploit the availability of a single class to define the subspace containing feature samples belonging to that specific class, have been exploited in [35, 44]. In more detail, such classifiers have been used only to compute the weights associated to each feature in [44], where the recognition process is then carried out computing Manhattan distances between weighted enrolment and authentication samples. A supervised classifier has been employed in [51] to estimate the performance achievable in open-set scenarios by dividing the available subjects into training and testing groups. Then for each user in the testing set, a specific k-NN classifier has been trained using samples from training subjects as impostors, and then evaluating the ability of rejecting unauthorized users considering the remaining subjects in the testing dataset. The available subjects have been divided into training and testing datasets also in [34], where a deep learning approach based on convolutional neural networks (CNNs) has been for the first time applied to KD data obtained from mobile devices. In more detail, the extracted features have been used as input to networks comprising either three or four unidimensional kernels, depending on the size of the employed PIN, with an additional fully connected layer and a softmax classifier. The parameters of the network have been learnt applying it to the training dataset with identification purposes, and then used to derive discriminative representations from the features referred to subjects in the testing dataset, evaluating on this latter the performance attainable in open-set scenarios.

5.2 | Free text comparison

User identification on the basis of free texts has been performed in [53]. As mentioned in Section 4.2, three different views, namely time series of alphabet keystrokes, special character keystrokes, and accelerometer values, have been there employed to represent the collected data. A deep learning approach based on recurrent neural networks has been then used to separately model each considered time series. Specifically, three gated recurrent unit bidirectional recurrent neural networks, trained with a RMSprop approach with Nesterov momentum, process the considered views to extract discriminative representations, with the last layers of each network concatenated before performing identification. Network training is performed for an increasing number of enrolled users, from only two to all the 26 available subjects, to evaluate the behaviour of the achievable accuracy. As expected, given well-known properties of identification processes, the achievable performance worsens as long as the number of considered user increases.

Standard machine learning approaches have been used in most of the studies performing closed-set verification on KD data acquired when typing free text on mobile devices. Specifically, RF has guaranteed the best recognition rates in [52, 55], while CART have been used together with LR in [11]. SVMs with Gaussian kernels have been employed in [56], and also neural networks have been exploited, as in [54, 61]. In more detail, a radial basis function network has been preferred over MLP in [61], since it requires the definition of only two parameters for each neuron. As mentioned in Section 4.2, five different networks have been trained, for varying sizes of the set with most recurrent characters. An MLP neural network with a single hidden layer and binary output has been instead employed in [54].

Open-set verification with free text has been evaluated only in [57], where Gaussian models have been employed to

characterize the behaviour of both key-specific and key-to-key transition features.

6 | PERFORMANCE

The best recognition rates achieved in the surveyed studies are reported in Table 3, grouped in categories defined on the basis of the employed input and the considered recognition modality. Specifically, the achieved results are expressed in terms of:

- False acceptance rate (FAR), providing the percentage of recognition attempts made by impostors that are falsely accepted in systems working in verification modality. Low values of FAR correspond to secure systems;
- False rejection rate (FRR), providing the percentage of recognition attempts made by legitimate users that are falsely rejected by the system, in systems working in verification modality. Low values of FRR correspond to usable systems;
- Equal error rate (EER), the operating point where FRR and FAR are equal. For a verification system to be accurate, the EER should be as low as possible;
- Correct identification rate, giving the probability of correctly determining the identity of the presented subject among a set of possible users, for systems working in identification modality.

The achieved results show that KD acquired through mobile devices is actually characterized by discriminative capabilities. Systems performing verification in open-set conditions, attaining EERs at about 2%–4%, have been in fact designed using both PINs and password as fixed-text inputs, as for instance in [25, 27, 31, 34, 36]. Even better results have been shown in [44], although evaluated on a quite small set of users. Thanks to be possibility of training classifiers on the whole set of subjects considered during testing, lower EERs have been achieved in closed-set verification conditions, such as in [26, 30], and [39]. Even though it represents a more challenging scenario, low error rates have been also achieved when processing free-text KD as in [52, 56], where closed-set conditions have been considered. The hardest scenario to consider seems the one involving OTPs, which actually represent inputs hard to be processed even when collected through computer keyboards [63].

However, it has to be remarked that a proper comparison of the recognition performance achieved in different studies cannot be carried out, not even considering works operating on the same kind of input and in the same modality, since distinct databases have been typically exploited for testing different approaches. The only proper comparisons can be done between [29] and [30] for PINs used in closed-set verification, among [33], [34], and [35] for PINs in open-set verification, as well as between [36] and [46] for logically strong passwords in open-set verification. In particular, the comparisons performed on PINs highlight the usefulness of exploiting deep learning approaches for KD-based recognition, with the methods in [30, 34] outperforming standard classifiers.

Other interesting insights on specific aspects can be derived from the analysis performed within selected studies. For instance, the feasibility of performing user recognition comparing KD samples acquired in distinct sessions has been explored in [25, 46]. Especially in the former paper, it has been observed that the actual variation of a subject's touch dynamics patterns cannot be captured if samples acquired from a single session are employed for enrolment purposes. The EERs obtained comparing KD samples in within-session and across-session conditions in fact notably increase, from 3.84% to 13.74% using the best classifier. Unfortunately, other databases containing KD samples captured in more than a single occasion have not been exploited to investigate this aspect, since data coming from different acquisition sessions have been employed for each user's enrolment.

The studies [25, 46], as well as [35], have also reported a detailed comparative analysis between the performance achievable in closed-set and open-set verification scenarios. Several approaches, learning discriminative properties for two-class and one-class problems, have been there exploited, with the former category notably outperforming the latter, as expected and shown by the results in Table 3.

Another interesting analysis, regarding the importance of posture, has been performed in [25], where it has been observed that a significant variability exists among the EERs achieved when using only a thumb, two thumbs, or an index to type on a mobile device. Furthermore, the authors have observed that entering a password in a system trained on a different posture may increase the attainable EER by up to 86%, with respect to assuming a fixed posture when the input is entered. It is therefore recommended, to improve the achievable recognition rates, to use posture-specific models when characterizing the KD collected from the considered users. Similar evidences and suggestions have been also reported in [56], in [11], and [57] where also the orientation of the employed mobile device has been taken into account, and in [39], where lying, sitting, standing, and walking postures have been considered. Also tests in [31] have highlighted the need for specific adaptations in the processing performed on data acquired in different postures, such as avoiding the accelerometer sensor due to its over-sensitivity, especially in the walking scenario.

The effects on recognition performance of varying the length of the employed inputs have been investigated in [31, 33, 34], and [35] for PINs, and in [40] for passwords. In general, an improvement in recognition rates has been always observed when enlarging the size of the employed entries. However, such gain may be significantly slow, and none of the aforementioned studies has considered multi-session scenarios, which reasonably may affect the observed behaviours, especially for longer inputs.

Moreover, several studies have compared the discriminative capabilities of different kinds of features. As already mentioned in Section 4.1, the higher discriminative capability of finger pressure and area, with respect to timing information,

has been for instance outlined in [25, 29, 33, 35, 36]. In more detail, finger pressure information has been typically reported as more relevant than finger area [28, 31]. *TC*, with related information such as drag, distance, and velocity, have been instead deemed more discriminative than finger pressure in [25, 51, 54, 57].

The effectiveness of including sensor information within the set of signals exploited to create KD representations has been remarked in [29, 53], where data from accelerometer provided the best recognition rates. The gyroscope has been observed to provide information more discriminative that touch location and accelerometer in [11, 37], and [36].

7 | OPEN ISSUES AND FURTHER RESEARCHES

Although biometric recognition on mobile devices based on KD has been investigated for more than a decade, it can be still considered a biometric modality at the early stages of its exploitation. Actually, the analysis of the typing patterns performed on mobile devices has attracted far less interest than its counterpart on physical keyboards, for which more in-depth evaluations have been conducted [64], and commercial applications have been proposed[15]. Nonetheless, given the pervasiveness of mobile devices, the rapid technology advancements which have led to the embedding of several miniaturized sensors, and the widespread of available applications, the design of systems exploiting KD to recognize the legitimate users of mobile devices would be an extremely interesting and profitable field of research. In fact, it would allow performing non-invasive and transparent access control to many services in our everyday life. However, before KD-based recognition can be considered a practical and feasible recognition modality for mobile devices, significant efforts have to be made and relevant aspects have to be still properly investigated.

One of the major issues affecting the current state of the art is the lack of public databases on which perform research. As evident from the information reported in Table 2 and in Section 6, most of the works proposed in literature have been tested on in-house databases, thus preventing the possibility of conducting proper comparison among different approaches, with only few exceptions [34, 36]. In order to foster the research on this topic, it is therefore of primary importance to collect, and make publicly available, datasets comprising KD recordings collected on mobile devices. In doing so, several aspects representing issues of currently available datasets have to be considered.

First of all, it has to be pointed out that the language analysed in most of researches on KD is English, which leaves the influence of the used language in the framework of biometric recognition still unexplored [65]. It would be therefore required to better investigate scenarios involving non-alphabetic characters to verify whether the proposed approach are robust to language changes.

It has then to be observed that, as shown in Table 2, too few works have been tested on databases comprising KD samples from a relevant number of subjects. The size of the employed database is an important parameter in evaluating the relevance and the reproducibility of the achieved results, especially for systems working in identification modality, whose performance depends on the number of considered users. Actually, acquisitions from several hundreds of subjects are commonly available in datasets used for testing other biometric modalities, including KD on hardware keyboards [64]. The only public datasets comprising KD recordings taken on mobile devices from more than 100 subjects are the ones described in [33] and [58]. Yet, while the former still has to be exploited to estimate achievable recognition performance, the latter one is characterized by a relevant issue affecting the reliability of the obtained results. Specifically, as for the most of in-house datasets mentioned in Table 2, the database in [33] has been collected performing a single recording session for each of the involved subjects. As already commented in Section 6, such condition cannot allow simulating a real-life behaviour, in which users may perform recognition attempts even at considerable temporal distances from enrolment, with the estimated recognition performance being therefore highly questionable. In addition to preventing a reliable evaluation of the recognition performance attainable in practical scenarios, the collection of single-session databases also hinders the possibility of investigating the permanence of discriminative characteristics in KD collected on mobile device, that is, the stability of the achievable recognition rates. In order to properly analyse this aspect, multiple recordings taken at an increasing temporal distance from the first one should be performed for each involved subject. Despite its importance, this aspect has never been taken into account when dealing with KD-based recognition on mobile devices. It is also worth mentioning that longitudinal experimental tests have instead been conducted on KD samples collected through computer keyboards, reporting sufficient stability of the traits acquired in sessions separated by time intervals in the order of months [64]. It would be therefore highly interesting to verifying this behaviour when typing on mobile devices too. The availability of longitudinal databases would also be useful to investigate the need for template update strategies, which have been recommended for biometric recognition systems based on KD collected through hardware keyboards [66]. Eventually, the dependence of the discriminative capabilities of the collected KD recordings on the age of the involved users should be analysed too.

Along with the need for comparing samples taken during different sessions, a peculiar aspect to be analysed about KD on mobile devices, much more relevant than when considering hardware keyboards, relates to the feasibility of recognizing a user through typing patterns recorded in different postures. The use of mobile devices in different conditions, typing with either a thumb, two thumbs, or an index finger, in either portrait or landscape orientations, while either walking, lying,

[15] https://www.behaviosec.com/

standing, or sitting, is in fact a common experience. As mentioned in Section 6, all the studies which have faced such scenarios agree on the need for developing specific models for each possible condition, in order to achieve good recognition rates. Yet, tests have been made so far on datasets comprising less than 40 subjects, and most of the time considering free text acquisitions, with the consequent need for devoting more efforts in analysing the issues deriving from posture.

Similarly to posture, even the emotional state of a subject can have an influence on the performed typing activity, and could be therefore taken into account when collecting and processing data [67]. Estimating the current mood of a person could in fact turn out to be beneficial to improve the achievable recognition rates, designing specific models to be exploited depending on the actual user's state of mind.

In addition to multi-session and multi-posture data, it would be also interesting to collect samples through multiple acquisition devices, as done in [45]. Such possibility would make it possible to evaluate the existence of device-independent characteristics within the typing behaviour of a user, or to investigate the feasibility of transferring the knowledge acquired while using a device into a different one, to improve the discriminative capability of models estimated on data collected through a single device. Cross-platform compatibility would notably ease the porting of services, while keeping the desired security, being the change of the mobile device usually more frequent than the change of the hardware keyboard. Furthermore, testing algorithms on samples collected on different devices would provide hints about which characteristics of the considered interfaces, such as height and width of the employed virtual keyboards and keys, have an effect on the achievable recognition performance.

Beyond simply operating on the size of the employed keyboards, it could also be worth taking more radical approaches, designing novel interfaces for mobile devices, able to improve usability and achieving better performance in KD-based recognition systems. For instance, a framework named "harmonized authentication based on thumbstrokes dynamics" (HATS), where a custom virtual keyboard based on a circular shape is adopted on a handheld device to let a user efficiently typing text employing only one thumb, has been proposed in [68]. According to the results there reported, the proposed HATS keyboard, when typing free text on mobile devices with both small and large touchscreens, seems guaranteeing both higher security and usability than standard QWERTY keyboards.

Further research is also required to properly address the dependency of the achievable recognition performance on the selected PINs or passwords. In addition to their lengths, several other characteristics of the employed inputs may in fact influence the recognition process. Differences in performance achievable with either easy, strong, or logically strong passwords have been in fact observed in [46], while a noticeable variability of the error rates attainable for distinct PINs having the same length has been reported in [31, 39]. The analysis of which characteristics result in better discriminability would be useful to provide suggestions during the password/PIN setup stages, thus increasing the security that KD-based recognition systems could guarantee. It is worth observing that, in order to carry out such analysis, it would be recommended to collect samples corresponding to several distinct PINs/passwords from each involved users. Due to the significant amount of time this kind of data collection process would require, such approach has been followed only in few works, such as [31, 40].

It would also be interesting to study the distinctiveness of the KD trait. For instance, the typing behaviours of only four out of the 12 subjects employed as legitimate users in [50] , and not those of the remaining eight participants, have shown high discriminative capability. Such outcome is particularly relevant given the fact that highly distinctive sensors such as accelerometer, gyroscope, and orientation have been employed in [50]. These observations may suggest that it would be worth designing techniques allowing users to perform proper enrolment on mobile devices, in order to make their KD more distinctive and thus guarantee better recognition performance. The feasibility of training subjects in typing has been recently investigated in [69], where instructions have been given to 24 participants, asked to enter passwords according to requests regarding specific aspects, such as the temporal gap to be taken between consecutive letters. Tests performed on recordings taken during two sessions a week apart have shown that it is actually possible to control typing features like flight time, hold time, and touch area, although certain characteristics are challenging to be modified. More in-depth investigations on these aspects would be beneficial for improving the typing skills of legitimate users, providing also the means to achieve diversity among the templates a user can employ in different applications. On the other side, malicious subjects could exploit the learnt lessons to launch more effective mimicry attacks, modifying their typing behaviour to make it resembling the one of the target user. In order to be able to deal with such possibility, samples of skilled forgeries from participants having the specific purpose of imitating characteristics of other users should be also included in the collected database. Such approach has been actually followed in [31], yet no significant improvements in FAR have been observed for attackers mimicking the observed behaviour, testifying the difficulty in copying specific typing patterns.

In order to improve the achievable recognition rates, the feasibility of integrating KD with other behavioural biometric traits, which could be easily captured by mobile devices without the need for any additional hardware, could also be investigated. It would be for instance interesting to design recognition systems exploiting, either separately or jointly, KD and swipe patterns [23]. These latter can be captured through the sensors available in touchscreens, whenever a user interact with a mobile phone to perform actions such as scrolling a page or providing a graphical password [70]. Although achieving worse recognition performance than KD, swipe patterns could be in fact useful to boost the recognition rates of a system jointly exploiting both modalities, as for example suggested in [48, 71]. Also the feasibility of integrating systems

based on KD with others exploiting 3D hand gesture performed with mobile devices as behavioural biometric modality might be worth to be investigated [72].

The recognition rates achievable exploiting KD on mobile devices could also be improved through a thoughtful application of deep learning approaches. Actually, to the best of our knowledge, a single work has evaluated so far the effectiveness of using CNNs to extract distinctive features from fixed-text inputs [34]. Recurrent networks have been instead considered for dealing with free text in [71], yet also in this case the potential of these approaches remains notably under-explored. Approaches employing generative models such autoencoders or generative adversarial networks have not been applied to KD collected on mobile devices. It has to be observed that, due to their limited length, it is not straightforward to efficiently process fixed-length inputs such as password and PINS with neural networks. However, given the remarkable recognition rates such methods have allowed to reach in several domains, it would be very interesting to investigate their applicability on KD data collected on mobile devices. It has however to be mentioned that, for a proper use of such methods, collecting large databases with a significant number of KD samples, recorded from many subjects, is of paramount importance, since otherwise the network training could be problematic.

Finally, it is worth remarking that the security of the collected data has not been taken into account in any of the works performing biometric recognition using KD on mobile devices. As it is widely known, biometric data can reveal several relevant information regarding their owners, which could be exploited for purposes other than biometric recognition, namely *function creep* [73]. For instance, KD could be exploited, to a limited extent, to monitor the health of a subject. Mood prediction of bipolar and non-bipolar persons has been performed in [74] analysing typing dynamics with convolutional and recurrent deep architectures. A similar evaluation has been also carried out in [75], where the feasibility of inferring useful information for patients with psychiatric disorders from their typing behaviour on mobile phones has been investigated. Fifteen emotional states, including confidence, hesitance, nervousness, relaxation, sadness, and tiredness, have been estimated from typing patterns in [67]. Also the educational level of a subject has been predicted based on collected KD in [76]. Furthermore, several studies have also estimated soft biometrics such as gender, age, and handedness of subjects whose KD have been recorded through smartphones [77]. The aforementioned studies testify that severe privacy issues may derive from the unauthorized access of malicious users to the samples collected while recording KD, or to the templates derived from it. Proper countermeasures, involving, for example, the usage of template protection schemes [73], have to be designed in order to implement privacy-compliant biometric recognition systems based on KD.

Privacy issues could also raise in case a system performing continuous recognition based on free-text acquisitions is designed, since the collected data could be employed to monitor the user level of attention and productivity. Although this possibility represents a more dangerous threat when dealing with systems employing KD collected through computer keyboards, such issues should however be properly considered and addressed before deploying KD-based systems for mobile devices in real-world applications.

8 | CONCLUSIONS

The state of the art regarding the use of KD collected on mobile devices has been surveyed in this paper. Specifically, an overview about the most significant studies which have been proposed on this topic, detailing the characteristics of the employed datasets, the used features, the adopted classification methods, and the achieved recognition performance, has been here provided. On the basis of the reviewed literature, several issues which still need to be properly addressed have been identified, and guidelines regarding possible future research lines have been reported.

Overall, according to the performed research, it is realistic that biometric recognition on mobile devices can be performed in real-life applications exploiting the peculiar characteristics of each user's typing patters. Being able to achieve such goal would be highly beneficial for integrating KD-based recognition systems into the access control mechanisms already employed when using mobile devices. Secure recognition methods would thus allow getting access to the devices themselves, as well as to services such as on-line payments, e-commerce, e-mail, or bank accounts, without requiring any specific action from the user. Nonetheless, several efforts need to be made before making the aforementioned scenarios applicable in real life, since many aspects affecting the recognition capabilities related to the typing patterns should still be properly explored, such as the effects of the adopted posture, the feasibility of cross-platform recognition, or the evaluation of the effectiveness of the chosen inputs.

REFERENCES

1. Jain, A., Ross, A., Nandakumar, K.: Introduction to Biometrics. Springer, New York (2011)
2. Sundararajan, A., Sarwat, A.I., Pons, A.: A survey on modality characteristics, performance evaluation metrics, and security for traditional and wearable biometric systems. ACM Comput. Surv. 52(2), 1–36 (2019)
3. Bleha, S., Slivinsky, C., Hussien, B.: Computer-access security systems using keystroke dynamics. IEEE Trans. Pattern Anal. Mach. Intell. 12, 1217–1222 (1990) Dec
4. Robinson, J.A., et al.: Computer user verification using login string keystroke dynamics. IEEE Trans. Syst. Man. Cybern. Syst. Hum. 28, 236–241 (1998) Mar
5. Jain, L., et al.: Passcode keystroke biometric performance on smartphone touchscreens is superior to that on hardware keyboards. International Journal of Research in Computer Applications & Information Technology IJCAIT. 2(4), 29–33 (2014)
6. GSMA Intelligence: The mobile economy 2020 march. https://www.gsma.com/mobileeconomy/wp-content/uploads/2020/03/GSMA_MobileEconomy2020_Global.pdf, 2020.Online (2020). Accessed 1 July 2020

7. Karlson, A., Brush, A., Schechter, S.: Can I borrow your phone?: Understanding concerns when sharing mobile phones. In: ACM Conference on Human Factors in Computing Systems (SIGCHI) (2009)
8. Meng, W., et al.: Surveying the development of biometric user authentication on mobile phones. IEEE Commun. Surv. Tutor. 17, 1268–1293 thirdquarter (2015)
9. Bryan, W.L., Harter, N.: Studies in the physiology and psychology of the telegraphic language. In: Psychological Review, 4, 1 27–53. (1897)
10. Gaines, R., et al.: Authentication by keystroke timing: some preliminary results Report R-2526-NSF. Rand Corporation, Santa Monica (1980)
11. Crawford, H., Ahmadzadeh, E.: Authentication on the go: Assessing the effect of movement on mobile device keystroke dynamics. In: Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017), pp. 163–173. USENIX Association, Santa Clara (2017)
12. Patel, V.M., et al.: Continuous user authentication on mobile devices: Recent progress and remaining challenges. IEEE Signal Process. Mag. 33, 49–61 (2016) July
13. Teh, P.S., et al.: A survey on touch dynamics authentication in mobile devices. Comput. Secur. 59, 210–235 (2016)
14. European Banking Authority: Opinion on the elements of strong customer authentication under PSD2 tech. rep. Online Accessed 1 July 2020 (2019)
15. Clarkeand, N., et al.: Keystroke dynamics on a mobile handset: a feasibility study. Inf. Manag. Comput. Secur. 11(4), 161–166 (2003)
16. Aviv, A.J., et al.: Practicality of accelerometer side channels on smartphones. In: Proceedings of the 28th Annual Computer Security Applications Conference, ACSAC '12, pp. 41–50. ACM, New York. (2012)
17. Alzubaidi, A., Kalita, J.: Authentication of smartphone users using behavioral biometrics. IEEE Commun. Surv. Tutor. 18, 1998–2026 thirdquarter (2016)
18. Banerjee, S.P., Woodard, D.L.: Biometric authentication and identification using keystroke dynamics: A survey. J. Pattern Recognit. Res. 7, 116–139 01 (2012)
19. Wobbrock, J.O.: Measures of text entry performance, *In: Text Entry Systems: Mobility, Accessibility, Universality*, Morgan Kaufmann, San Francisco (2007)
20. Zhang, M.R., Zhai, S., Wobbrock, J.O.: Text entry throughput: Towards unifying speed and accuracy in a single performance metric. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI '19, pp. 636:1–636:13. ACM, New York (2019)
21. Stanciu, V.-D., et al.: On the effectiveness of sensor-enhanced keystroke dynamics against statistical attacks. In: Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy, CODASPY '16, pp. 105–112. ACM (2016)
22. Goel, M., et al.: Gripsense: Using built-in sensors to detect hand posture and pressure on commodity mobile phones. In: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, UIST '12, pp. 545–554. ACM, New York (2012)
23. Frank, M., et al.: Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. IEEE Trans. Inf. Forensics Secur. 8, 136–148 (2013) Jan
24. Saevanee, H., Bhattarakosol, P.: Authenticating user using keystroke dynamics and finger pressure. In: IEEE Consumer Communications and Networking Conference (2009)
25. Buschek, D., De Luca, A., Alt, F.: Improving accuracy, applicability and usability of keystroke biometrics on mobile touchscreen devices. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15, pp. 1393–1402. ACM, New York (2015)
26. Zahid, S., et al.: Keystroke-based user identification on smart phones. In: International Workshop on Recent Advances in Intrusion Detection (RAID) (2009)
27. Hwang, S.-S., Cho, S., Park, S.: Keystroke dynamics-based authentication for mobile devices. COSE. 28, 85–93 (2009)
28. Tasia, C.-J., et al.: Two novel biometric features in keystroke dynamics authentication systems for touch screen devices. In: Proceedings of Security and Communication Networks, no. 7, pp. 750–758. John Wiley and Sons, Ltd.Wiley Online Library, Changhua City (2013)
29. Ho, G.: TapDynamics: strengthening user authentication on mobile phones with keystroke dynamics, Technical report, Stanford University (2013)
30. Deng, Y., Zhong, Y.: Keystroke dynamics advances for mobile devices using deep neural network. In: Recent Advances in User Authentication Using Keystroke Dynamics Biometrics. Science Gate Publishing, Xanthi (2015)
31. Feng, T., et al.: You are how you touch: User verification on smartphones via tapping behaviors. In: IEEE International Conference on Network Protocols (2014)
32. de Mendizabal-VÃ¡zquez, I., et al.: Supervised classification methods applied to keystroke dynamics through mobile devices. In IEEE International Carnahan Conference on Security Technology ICCST) (2014)
33. Teh, P. S., et al.: Strengthen user authentication on mobile devices by using user's touch dynamics pattern. J Ambient Intell Human Comput 11, 4019–4039 (2020)
34. Maiorana, E., Kalita, H., Campisi, P.: Deepkey: Keystroke dynamics and CNN for biometric recognition on mobile devices. In: European Workshop on Visual Information Processing (EUVIP) (2019)
35. Teh, P., et al.: Strengthen user authentication on mobile devices by using user's touch dynamics pattern. J Ambient Intell Human Comput 11, 4019–4019 (2019)
36. Kalita, H., Maiorana, E., Campisi, P.: Keystroke dynamics for bsiometric recognition in handheld devices. In: IEEE International Conference on Telecommunications and Signal Processing (TSP) (2019)
37. Coakley, M.J., Monaco, J.V., Tappert, C.C.: Keystroke biometric studies with short numeric input on smartphones. In: 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS), pp. 1–6. (2016)
38. Lee, H., et al.: A parameterized model to select discriminating features on keystroke dynamics authentication on smartphones. Pervasive Mob. Comput. 54, 45–57 (2019)
39. Wang, Y., et al.: Improving reliability: User authentication on smartphones using keystroke biometrics. IEEE Access 7, 26218–26228 (2019)
40. Campisi, P., et al.: User authentication using keystroke dynamics for cellular phones. *IET Signal Processing* 3(4), 333–341 (2009)
41. Antal, M., Szabó, L.Z., László, I.: Keystroke dynamics on android platform. In: Proceedings of International Conference Interdisciplinarity in Engineering INTER-ENG 2014, 8th, edn., pp. 820–826. Elsevier, Tirgu-Mures (2014)
42. Alshanketi, F., Traore, I., Ahmed, A.A.: Improving performance and usability in mobile keystroke dynamic biometric authentication. In: 2016 IEEE Security and Privacy Workshops (SPW), pp. 66–73. (2016) May
43. El-Abed, M., Dafer, M., Khayat, R.E.: Rhu keystroke: A mobile-based benchmark for keystroke dynamics systems. In: 2014 International Carnahan Conference on Security Technology (ICCST), pp. 1–4. (2014)
44. Giuffrida, C., et al.: I sensed it was you: Authenticating mobile users with sensor-enhanced keystroke dynamics. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA) (2014)
45. Al-Obaidi, N., Al-Jarrah, M.: Statistical keystroke dynamics system on mobile devices for experimental data collection and user authentication. In: International Conference on Developments in eSystems Engineering (2016)
46. Antal, M., Nemes, L.: The mobikey keystroke dynamics password database: Benchmark results. In: Silhavy, R., Senkerik, R., Oplatkova, Z.K., Silhavy, P., Prokopova, Z. (eds.), Software Engineering Perspectives and Application in Intelligent Systems, pp. 35–46. Springer International Publishing, Cham (2016)
47. El-Abed, M., Dafer, M., Rosenberger, C.: Rhu keystroke touchscreen benchmark. In: Proceedings of International Conference on Cyberworlds. IEEE, Caen (2018)
48. Tse, K., Hung, K.: User behavioral biometrics identification on mobile platform using multimodal fusion of keystroke and swipe dynamics and recurrent neural network. In: IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE) (2020)

49. Trojahn, M., Arndt, F., Ortmeier, F.: Authentication with keystroke dynamics on touchscreen keypads - effect of different n-graph combinations. In: International Conference on Mobile Services, Resources, and Users (MOBILITY) (2013)
50. Gascon, H., et al.: Continuous authentication on mobile devices by analysis of typing motion behavior 03 (2014)
51. Kambourakis, G., et al.: Introducing touchstroke: keystroke-based authentication system for smartphones. Secur. Commun. Network. 9, 542554 (2014)
52. Feng, T., et al.: Continuous mobile authentication using virtual key typing biometrics. In: International Conference on Trust, Security and Privacy in Computing and Communications (2013)
53. Sun, L., et Sequential keystroke behavioral biometrics for mobile user identification via multi-view deep learning. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer International Publishing (2017)
54. Inguanez, F., Ahmadi, S.: Securing smartphones via typing heat maps. In: International Conference on Consumer Electronics-Berlin (2017)
55. Alshanketi, F., Traoré, I., Awad, A.: Multimodal mobile keystroke dynamics biometrics combining fixed and variable passwords. Secur. Priv. 2(1), 1–14 (2019)
56. Cilia, D., Inguanez, F.: Multi-model authentication using keystroke dynamics for smartphones. In: International Conference on Consumer Electronics-Berlin (2018)
57. Buschek, D., et al.: Researchime: A mobile keyboard application for studying free typing behaviour in the wild. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18, pp. 255:1–255:14. ACM, New York (2018)
58. Belman, A., et al.: Insights from BB-MAS - a large dataset for typing, gait and swipes of the same person on desktop, tablet and phone. In: arXiv prepring (2019)
59. Jokinen, J.P.P., et al.: Modelling learning of new keyboard layouts. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, pp. 4203–4215. ACM, New York (2017)
60. Alghamdi, S.J., Elrefaei, L.A.: Dynamic user verification using touch keystroke based on medians vector proximity. In: Proceedings of International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), 7th edn., pp. 56–61. IEEE (2015)
61. Jain, L., et al.: Authenticating mobile phone users using keystroke analysis. Int. J. Inf. Secur. 6, 1–14 (2007)
62. Frank, E., Hall, M.A., Witten, I.H.: The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2016)
63. Bours, P., Masoudian, E.: Applying keystroke dynamics on one-time PIN codes. In: International Workshop on Biometrics and Forensics (2014)
64. Morales, A., et al.: KBOC: Keystroke biometrics ongoing competition. In: IEEE International Conference on Biometrics: Theory, Applications, and Systems (2016)
65. Samura, T., Izumi, M., Nishimura, H.: Flick input authentication in japanese free text entry on smartphones. In: 2014 Proceedings of the SICE Annual Conference (SICE), pp. 1348–1353. (2014)
66. Giot, R., Dorizzi, B., Rosenberger, C.: Analysis of template update strategies for keystroke dynamics. In: IEEE Workshop on Computational Intelligence in Biometrics and Identity Management (CIBIM) (2011)
67. Epp, C., Lippold, M., Mandryk, R.: Identifying emotional states using keystroke dynamics. In: SIGCHI (2011)
68. Zhou, L., et al.: Harmonized authentication based on thumbstroke dynamics on touch screen mobile phones. Decis. Support Syst. 92, 14–24 (2016)
69. Mecke, L., et al.: Exploring intentional behaviour modifications for password typing on mobile touchscreen devices. In: Proceedings of the Fifteenth Symposium on Usable Privacy and Security (SOUPS) (2019)
70. De Luca, A., et al.: Touch me once and i know it's you! implicit authentication based on touch screen patterns. In: International conference on Human factors in computing systems (CHI) (2012)
71. Tse, K., Hung, K.: Behavioral biometrics scheme with keystroke and swipe dynamics for user authentication on mobile platform. In: 2019 IEEE 9th Symposium on Computer Applications Industrial Electronics (ISCAIE) (2019)
72. Guerra-Casanova, J., et al.: Authentication in mobile devices through hand gesture recognition. Int. J. Inf. Secur. 11, 65–83 (2012)
73. Nandakumar, K., Jain, A.K.: Biometric template protection: Bridging the performance gap between theory and practice. IEEE Signal Process. Mag. 32(5), 88–100 (2015)
74. Huang, H., et al.: dpmood: Exploiting local and periodic typing dynamics for personalized mood prediction. In: 2018 IEEE International Conference on Data Mining (ICDM) (2018)
75. Cao, B., et al.: Modeling mobile phone typing dynamics for mood detection. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17 (2017)
76. Tsimperidis, I., et al.: R2bn: An adaptive model for keystroke-dynamics-based educational level classification. IEEE Trans. Cybern. 50(2), 525–535 (2020)
77. Buriro, A., et al.: Age, gender and operating-hand estimation on smart mobile devices. In: IEEE BIOSIG (2016)

How to cite this article: Maiorana E, Kalita H, Campisi P. Mobile keystroke dynamics for biometric recognition: An overview. *IET Biome*. 2020;1–23. https://doi.org/10.1049/bme2.12003