Type2Branch: Keystroke Biometrics based on a Dual-branch Architecture with Attention Mechanisms and Set2set Loss

Nahuel González*, Giuseppe Stragapede†, Ruben Vera-Rodriguez†, Ruben Tolosana†
*Laboratorio de Sistemas de Informacion Avanzados (LSIA), University of Buenos Aires, Argentina
†Biometrics and Data Pattern Analytics (BiDA) Lab, Universidad Autonoma de Madrid, Spain

***Abstract*—In 2021, the pioneering work on TypeNet showed that keystroke dynamics verification could scale to hundreds of thousands of users with minimal performance degradation. Recently, the KVC-onGoing competition[1] has provided an open and robust experimental protocol for evaluating keystroke dynamics verification systems of such scale, including considerations of algorithmic fairness. This article describes Type2Branch, the model and techniques that achieved the lowest error rates at the KVC-onGoing, in both desktop and mobile scenarios. The novelty aspects of the proposed Type2Branch include: *i)* synthesized timing features emphasizing user behavior deviation from the general population, *ii)* a dual-branch architecture combining recurrent and convolutional paths with various attention mechanisms, *iii)* a new loss function named Set2set that captures the global structure of the embedding space, and *iv)* a training curriculum of increasing difficulty. Considering five enrollment samples per subject of approximately 50 characters typed, the proposed Type2Branch achieves state-of-the-art performance with mean per-subject EERs of 0.77% and 1.03% on evaluation sets of respectively 15,000 and 5,000 subjects for desktop and mobile scenarios. With a uniform global threshold for all subjects, the EERs are 3.25% for desktop and 3.61% for mobile, outperforming previous approaches by a significant margin.**

***Index Terms*—Type2Branch, Set2set loss, keystroke dynamics, behavioral biometrics, synthetic data, security**

I. Introduction

KEYSTROKE Dynamics (KD) refers to the typing behavior exhibited by human subjects, and it represents a form of *behavioral biometric* trait, similar to gait [1], touch gestures [2], [3], and signature [4], among others. In its most basic form, keystroke dynamics is captured as discrete time events: the times at which a key is pressed and released (typically in Unix time format), along with the corresponding key code (ASCII). Additional information, such as key pressure or fingertip size, may be available based on specific hardware capabilities, for example as in BehavePassDB [5]. Consequently, applications based on keystroke dynamics are generally cost-effective, as they only require standard keyboards, which currently serve as the primary means for inputting textual data into digital systems, utilized by billions of users daily.

Behavioral biometrics, such as KD, currently do not achieve the same recognition accuracy as their *physiological* counterparts, such as face, fingerprint, and iris. Nevertheless, they offer the advantage of operating *transparently*, without requiring the subject to perform any specific procedure. Typically, biometric recognition-based security is the most prevalent application of KD. This involves both operational modes: *i)* verification: pairs of KD samples are captured, processed, and matched while subjects engage in activities like writing an email or taking a test on educational platforms. It can also serve as an additional layer of biometric security alongside traditional knowledge-based passwords [6]; and *ii)* identification: KD enables linking different accounts used by the same individual by matching their typing behavior among multiple samples from other subjects. Identifying or shortlisting malicious users can contribute to digital forensics applications, such as tackling toxicity, hate, and harassment on social networks [7], protecting children from online grooming [8], and combating the spread of fake news [9] and "Wikipedia wars" [10].

Additionally, as other biometric characteristics that exhibit patterns associated with demographic groups [11], [12], KD studies have assessed predictability in gender [13], age [14], emotions [15], and even mother tongue [16], representing an unexplored, yet stimulating, field of research [17].

KD can be broadly categorized based on two criteria: *i)* the type of acquisition device (keyboard), distinguishing between desktop and mobile. Mobile touchscreens tend to exhibit more variability due to differences in pose or typing activity compared to desktop keyboards; and *ii)* concerning the text format, it can be classified as free, fixed, or transcript. In the case of free text, variations exist across different samples, resulting in sparser, less structured data with a higher incidence of typing errors. In contrast, fixed-text scenarios, such as an intruder typing a victims password, aim for consistent representation and lower error rates. Finally, transcript text is considered a hybrid format, involving subjects reading, memorizing, and typing a presented text. It is important to note that composition (free text) and transcription tasks produce equivalent evaluation results [18] when used for training. As transcription is easier for the subjects, current large keystroke dynamics datasets like the one used in this study have almost exclusively adopted this modality for enrollment.

In this scenario, we propose a new approach for transcript text KD-based biometric verification called Type2Branch. Fig. 1 shows a graphical representation of the workflow of Type2Branch. Type2Branch proposes several novelty as-

Email: ngonzalez@lsia.fi.uba.com
[1]https://sites.google.com/view/bida-kvc/

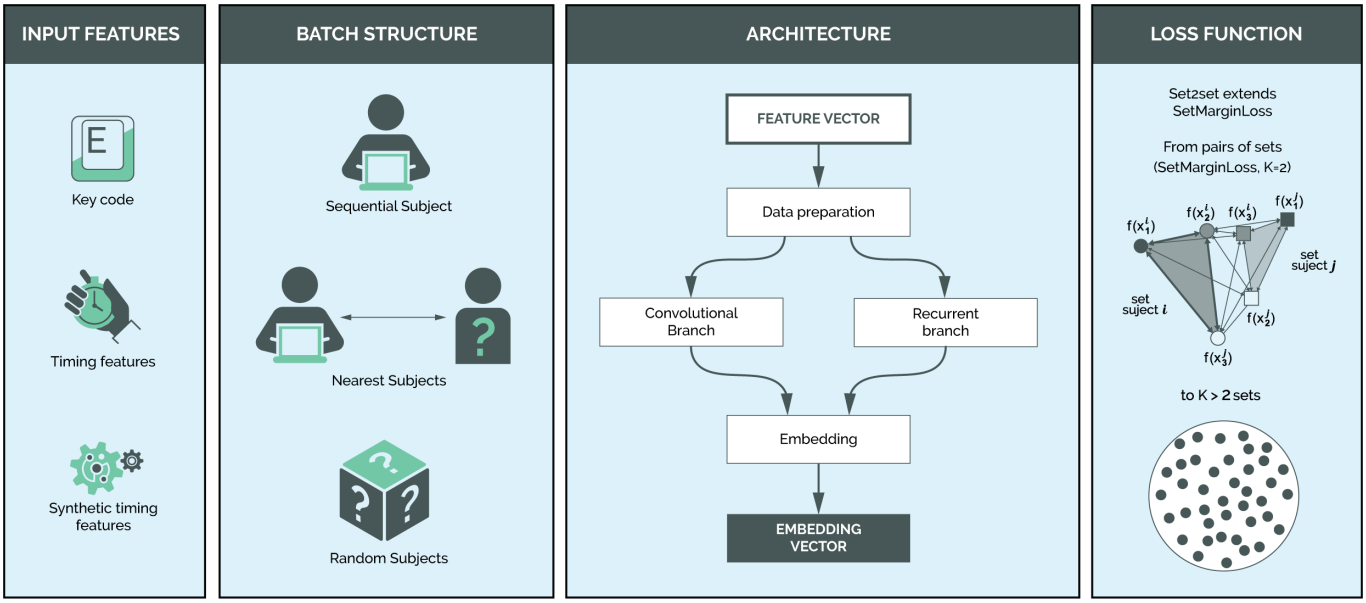


Fig. 1: Graphical representation of the workflow of Type2Branch, the proposed biometric keystroke verification system.

pects that contribute to achieving a significant improvement in the verification performance in comparison with existing approaches in the literature. They can be listed as follows:

- We propose a two-branch model architecture with self-attention modules, starting from the observation that keystroke timings result from a combination of two factors: a partially conscious decision process involving *what* to type and an entirely unconscious motor process pertaining to *how* to type [19]. The convolutional branch is expected to excel at identifying common, short sequences, while the recurrent branch is expected to capture the user's time-dependent decision process.
- Previous Distance Metric Loss (DML) approaches such as the SetMargin loss proposed by Morales *et al.* [20] extend the triplet loss by considering pairs of sets of samples instead of triplets. In this work, we propose the Set2set loss, which extends this idea by considering K sets at a time. This allows the model to map much more effectively the latent hyperspace, leading to improved recognition performance. As detailed in Sec. III, we use an optimized implementation of the proposed Set2set loss for computing speed, given that a naïve implementation of the deeply nested loop implicit in its formulation is prohibitively slow even for small batches.
- Due to the recent popularity of synthetic data to overcome challenges in biometrics [21], [22], we propose to extract synthetic timing features from the general population profile as part of the learning framework in order to allow the model to learn more subject-specific features. The synthetic features are generated with the tool reported at [23], in which source code are publicly available.
- We designed a learning curriculum of increasing difficulty in order to progressively show to the network the nearest, i.e. harder to discriminate, users while at the same time still including enough random sets for the model not to lose track of the global structure of the embedding space.

In light of these, Type2Branch improves on the performance of state-of-the-art models in the Keystroke Verification Challenge - onGoing (KVC-onGoing)[2] [17], [24], with 3.25% global Equal Error Rate (EER) in the desktop task, and 3.61% global EER in the mobile task, corresponding to the first place in both tasks. Considering the mean per-subject distribution (see Sec. V-C), the EERs achieved by Type2Branch are further reduced to 0.77% and 1.03%, respectively. KVC was launched to provide a public and reproducible way to benchmark KD-based verification systems in desktop and mobile scenarios, using large-scale databases (over 185,000 subjects in total) and a standard experimental protocol. In addition, not only the verification performance is considered in KVC, but also the demographic fairness and privacy aspects of the biometric systems. The KVC-onGoing extends its limited-time edition within the 2023 IEEE International Conference on Big Data.

The remainder of the article is organized as follows: first, an overview of previous related works is included in Sec. II. Then, Sec. III provides a detailed presentation of all aspects of the proposed Type2Branch. Sec. IV presents the datasets adopted, while the experimental protocol is illustrated in Sec. V. Finally, Sec. VI and Sec. VII respectively contain the analysis of the results obtained and the article conclusive remarks.

II. Related Work

A. Keystroke Dynamics for Biometric Verification

The idea of classifying subjects based on their typing behavior dates back to the introduction of personal computers. In the early days of keystroke dynamics, only mechanical (desktop) keyboards were available, for which the literature is more extensive in comparison with the more recent mobile settings related to touchscreen devices. Additionally, the processing

[2]https://sites.google.com/view/bida-kvc/

power and machine learning algorithms at the disposal of the researchers were not nearly comparable to today's scenario. In fact, in biometrics, deep learning-based approaches have dramatically increased the recognition performance in comparison with hand-crafted algorithms [25]. For a complete literature review about KD, we invite the reader to consult [26], [27]. In the remainder of this section, the latest developments based on deep learning approaches, which are most relevant to the current research work, will be presented.

In [28], it was demonstrated that a deep neural network improved the performance of hand-crafted algorithms when evaluated on the CMU database [29]. Neural network-based approaches were also used for auxiliary tasks aimed at enhancing authentication performance, such as predicting digraphs absent from enrollment samples by analyzing the relationships between keystrokes [30]. A Convolutional Neural Network (CNN) coupled with Gaussian data augmentation for the fixed-text scenario was introduced in [31], while a neural network was applied to RGB histograms derived from fixed-text keystroke in [32]. Additionally, Multi-Layer Perceptron (MLP) architectures have been investigated [33]. In [34], a combination of convolutional and recurrent neural networks (RNNs) was designed to extract higher-level keystroke features from the SUNY Buffalo database [35]. The convolutional process precedes feeding the sequence into the recurrent network to better characterize the keystroke sequence. RNN variants are widely used in keystroke biometrics, as seen in [36] (bidirectional RNN), or in [37], where keystroke sequences are structured as image-like matrices and processed by a CNN combined with a Gated Recurrent Unit (GRU) network.

Generally speaking, the proliferation of machine learning algorithms capable of analyzing and learning human behaviors thrives on large-scale datasets. To this end, the Aalto databases, proposed in two popular Human-Computer Interaction (HCI) studies on people's typing behavior on desktop [38] and mobile devices [39], are extremely useful. These databases were collected by the User Interfaces[3] group of Aalto University (Finland). The desktop[4] [38] database comprises around 168,000 subjects, while the mobile[5] [39] one encompasses approximately 60,000 subjects. As can be seen in Table I, the size of such databases is significantly greater than other public databases, effectively reflecting the challenges associated with current massive application usage.

The work of Acien *et al.* [25] is among the first studies that adopted Aalto databases [38] for biometrics purposes. In addition to this, a novelty aspect of their work is the introduction of Long Short-Term Memory (LSTM) Recurrent Neural Networks (RNNs) trained with triplet loss [46], that are employed with an *off-line* approach (i.e. a common recognition model is trained to extract salient features in the user's biometric traits on a dataset before being deployed for user classification). They analyzed to what extent deep learning models are able to scale in keystroke biometrics to recognize users from a large pool while attempting to minimize the amount of data per user required for enrollment. This system, called TypeNet, was able to verify subjects' identities when the amount of data per user is scarce, using only 5 enrollment samples and 1 test sample per user, with 50 characters typed per sample. TypeNet maps input data into a learned representation space that reveals a "semantic" structure based on distances. Such approach is known as Distance Metric Learning (DML) method. In [20], a novel DML method, called SetMargin loss (SM-L), was proposed to address the challenges associated with transcript-text keystroke biometrics in which the classes used in learning and inference are disjoint. SM-L is based on a learning process guided by pairs of sets instead of pairs of samples, as contrastive or triplet loss consider, allowing to enlarge inter-class distances while maintaining the intra-class structure of keystroke dynamics. This led to improved recognition perfomance with TypeNet.

Later on, replicating the same experimental protocol as [25], Stragapede *et al.* proposed TypeFormer [47]. TypeFormer was developed starting from the Transformer model [48], with several adaptations to optimize its recognition performance for KD. The model consists of Temporal and Channel Modules enclosing two Long Short-Term Memory (LSTM) recurrent layers, Gaussian Range Encoding (GRE), a multi-head Self-Attention mechanism, and a Block-Recurrent structure. In several experiments, TypeFormer outperformed TypeNet in the mobile environment, but not in the desktop case [17].

Recently, in [49], a novel approach called DoubleStrokeNet for recognizing subjects using bigram embeddings was proposed. This is achieved using a Transformer-based neural network that distinguishes between different bigrams. Additionally, self-supervised learning techniques are used to compute embeddings for both bigrams and users. The authors experimented with the Aalto databases, reaching very competitive results in terms of recognition performance. In such cases, while the ideas presented are very interesting, it is often difficult to compare across different studies, which adopt

TABLE I: Some of the most important public keystroke dynamics databases in chronological order.

Database	Scenario	No. of Subjects	Text Format	Strokes per Subject
GREYC (2009) [40]	Desktop	133	Fixed	∼800
CMU (2009) [29]	Desktop	51	Fixed	∼400
BiosecurID (2010) [41]	Desktop	400	Free	∼200
RHU (2014) [42]	Desktop	53	Fixed	∼600
Clarkson I (2014) [43]	Desktop	39	Fixed, free	∼20k
SUNY (2016) [35]	Desktop	157	Transcript, free	∼17k
Clarkson II (2017) [44]	Desktop	103	Free	∼125k
Aalto Desktop (2018) [38]	**Desktop**	**168k**	**Transcript**	**∼750**
Aalto Mobile (2019) [39]	**Mobile**	**37k**	**Transcript**	**∼750**
HuMIdb (2020) [45]	Mobile	600	Fixed	∼20
BehavePassDB (2022) [5]	Mobile	81	Free	∼100

[3]https://userinterfaces.aalto.fi/

[4]https://userinterfaces.aalto.fi/136Mkeystrokes/

[5]https://userinterfaces.aalto.fi/typing37k/

different experimental settings. To this end, the first attempt to promote reproducible research and establish a baseline in biometric recognition using keystroke biometrics was carried out in 2016 in the form of a competition for KD by Morales *et al.* [6], namely Keystroke Biometrics Ongoing Competition (KBOC). A total of 12 institutions from 7 different countries registered for the competition. In that case, the dataset used consisted of keystroke sequences (fixed text) from 300 users acquired in 4 different sessions.

Following this line of research, the Keystroke Verification Challenge - onGoing (KVC-onGoing)[6] was recently launched, considering both Aalto databases (the desktop database was used for Task 1, the mobile one for Task 2). A limited-time edition of the challenge was held within the 2023 IEEE International Conference on Big Data (IEEE BigData)[7], which is now made ongoing. The challenge is hosted on CodaLab[8]. Thanks to the demographic (age, gender) labels present in the original database, a study of the demographic differentials in the scores was also carried out for purposes such as privacy quantification and fairness, alongside a thorough evaluation of the biometric verification performance. TypeNet and TypeFormer were also benchmarked on the KVC, which attracted the participation of 7 teams. Some of the teams were able to outperform both TypeNet and TypeFormer on both scenarios. In particular, the approach presented in this paper, Type2Branch, holds the first position in both desktop and mobile tasks in the KVC-onGoing. For more information about the details of the competition, we invite the reader to consult [17], [24].

B. Generation of Synthetic Features

KD verification systems have traditionally been assessed under a zero-effort attack model. In other words, biometric samples captured from different subjects are compared, but no effort is made to emulate the characteristics of genuine subjects. To this end, recent studies have demonstrated that attacks employing statistical models and synthetic forgeries can yield significant success rates [50], [51], raising concerns that zero-effort approaches are overly optimistic.

In [52], the authors explored spoofing techniques leveraging higher-order contexts and empirical distributions to generate artificial samples of keystroke timings to improve existing attacks. Additionally, they proposed a new general method for the detection of synthetic forgeries to protect against sample-level attacks, at the cost of a small penalty in overall accuracy. A comprehensive evaluation of the proposed detection and spoofing methods was carried out in two scenarios: the attacker having access to all the legitimate user samples, and the attacker having access to general population data only. One of the proposed spoofing methods was able to double and sometimes triple the false acceptance rates.

Following this line of research, in [53], the synthesis of KD features was achieved based on universal, user-dependent, and generative models. The synthetic features were used to improve the training process of keystroke-based bot detection systems. In their performance analysis, the authors considered several aspects such as the amount of data available to train the bot detector, the type of synthetic data used to model the human behavior, and the input text dependencies.

In our proposed Type2Branch, the generation of synthetic data is incorporated as part of the learning framework to represent the average typing behavior of the entire training population, with the objective of reflecting how the behavior of each subject shows distinguishable patterns in comparison with the population profile. We adopt the implementation presented in [23].

III. Type2Branch: Proposed System

Fig. 1 shows a graphical representation of the workflow of Type2Branch. We describe next the key modules of our proposed keystroke verification system.

A. Terminology and Definitions

A keystroke dynamics sample $\mathbf{w}$ is a sequence $\mathbf{w}_1, \ldots, \mathbf{w}_M$, of fixed length M, of tuples of the form

$$(k_i,\, t_i^P,\, t_i^R)$$

where k_i is the integer key code corresponding to the i–th keystroke, t_i^P is the timestamp of its key press event, and t_i^R is the timestamp of its key release event.

A class is given by the set of samples of a single subject. We assume that the number N of samples in each class, is fixed. A batch consists of NK samples, where all the N samples of K selected classes are included. The n–th sample of the k–th class is denoted as $\mathbf{w}_n^k$ and its corresponding embedding vector as $\mathbf{x}_n^k$.

The center of the embeddings for the k–th set is denoted as $\mu(k)$ and is calculated in the form

$$\mu(k) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i^k \tag{1}$$

The mean radius $r(k)$ of the k–th set embeddings cluster is given by

$$r(k) = \frac{1}{N}\sum_{i=1}^{N} d(\mathbf{x}_k^i, \mu(k)) \tag{2}$$

Using the above definition, the *radius penalty* for a batch is defined as

$$\mathcal{L}_{RP} = \frac{1}{K}\sum_{k=1}^{K}\left|\frac{r(k)}{R} - 1\right| \tag{3}$$

where the term R, defined as

$$\mathrm{R} = \frac{1}{K}\sum_{k=1}^{K} r(k) \tag{4}$$

is the mean radius over all classes.

[6]https://sites.google.com/view/bida-kvc/
[7]https://bigdataieee.org/BigData2023/
[8]https://codalab.lisn.upsaclay.fr/competitions/14063

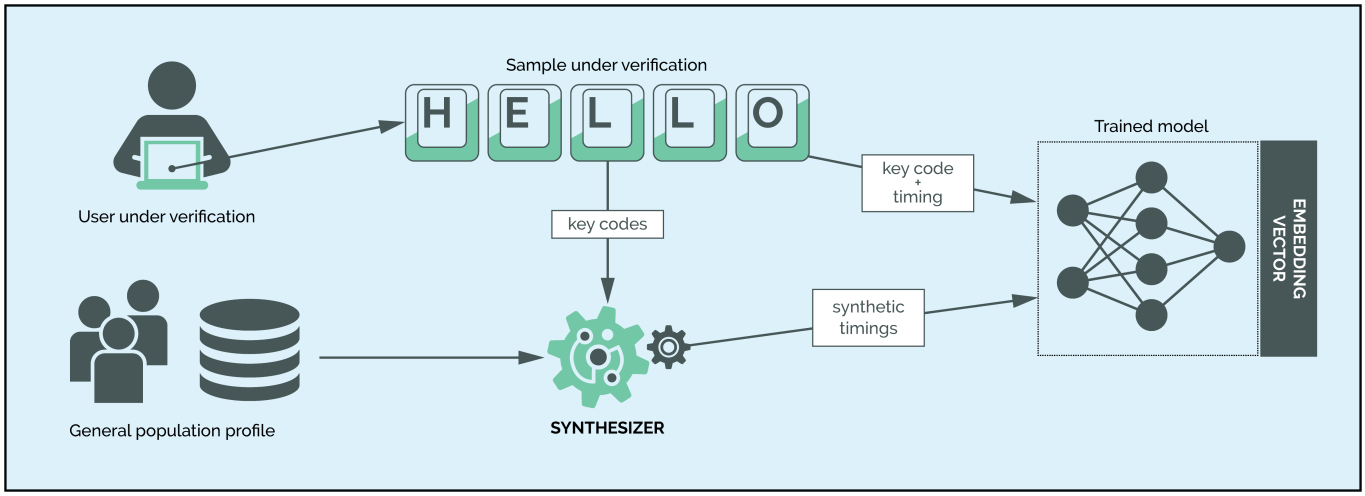


Fig. 2: Representation of the feature extraction process. The key code sequence is used to generate synthetic timings based on the general population profile, which are considered together with the original keystroke timings.

B. Input Features

Fig. 2 shows a representation of the feature extraction process. The key code sequence of the sample is used to generate synthetic timings; for this purpose, a general population profile trained with all available samples is used. The synthetic timings, together with the original key codes and keystroke timings, are used as input features.

In total, five input features per keystroke are used to train the model: VK, the integer key code; two classical timing features, and two synthetic timing features. The classic timing features are HT, the hold time (interval between key press and release events), and FT, the flight time (interval between the previous and current key press events). The corresponding synthetic timing features, SHT and SFT, are used to reflect how the typing style of the subject differs from that of the general population. All timing features are scaled to seconds and clipped to a maximum value of 10.

The two classic input features, hold time (HT) and flight time (FT), are calculated as

$$t_i^{HT} = t_i^R - t_i^P \tag{5}$$

$$t_i^{FT} = t_i^P - t_{i-1}^P \quad \text{for } i > 0, \text{ or } 0 \text{ otherwise} \tag{6}$$

The synthetic features are calculated using the finite context modelling method [52], [54], here denoted by $\mathcal{S}$. Given a target sample $\mathbf{w}$, the method $\mathcal{S}$ outputs a new sample $\mathbf{w}^S$ with the same keystroke sequence but synthesized hold times and flight times. For this purpose, it requires a profile $\mathcal{A}$, which consists of a set of samples that represent the behavior to be synthesized; whether that of a specific subject, a group of subjects, or, in the scope of this study, that of the general population as a whole. Symbolically,

$$\mathbf{w}^S = \mathcal{S}(\mathcal{A}, \mathbf{w}) \tag{7}$$

where $\mathbf{w}^S$, in the same way as $\mathbf{w}$, is also a sequence $\mathbf{w_1^S}, \ldots, \mathbf{w_n^S}$ of length n of tuples of the form

$$(k_i,\, t_i^{SP},\, t_i^{SR}) \tag{8}$$

Briefly, the finite context modelling method attempts to match, for each key, short keystroke sequences that precede it in the sample $\mathbf{w}$ with similar sequences that can be found in the profile $\mathcal{A}$. In this way, statistical distributions for each keystroke timing are inferred, which can then be sampled to output the final synthesized timings [52], [54].

Here, the profile $\mathcal{A}$ used to generate the synthetic input features collects all the samples in the development set, with the objective of representing the average typing behavior of the entire training population. For each sample $\mathbf{w}$, the corresponding $\mathbf{w}^S$ is generated using the tool [23] and the two synthetic features are calculated as

$$s_i^{HT} = t_i^{HT} - (t_i^{SR} - t_i^{SP}) \tag{9}$$

$$s_i^{FT} = t_i^{FT} - (t_i^{SP} - t_{i-1}^{SP}) \tag{10}$$

The resulting feature vector for each keystroke is thus

$$(k_i,\, t_i^{HT},\, t_i^{FT},\, s_i^{HT},\, s_i^{FT}) \tag{11}$$

which is the input the model receives during training, validation, and evaluation.

C. Batch structure

The number N of samples per subject and the number K of subjects per batch is fixed, for a total of NK samples per batch. During epoch zero, the K subjects to be included in each batch are randomly chosen from all those available in the development set, with reposition between batches but making sure no batch includes repeated subjects.

For epochs $m > 0$, the objective of the training curriculum is to progressively show to the model the nearest, i.e. harder to discriminate, subjects while at the same time still including enough random sets for the model not to lose track of the global structure of the embedding space. For this reason, each batch includes:

- *One sequential subject*, in the same order as they appear in the development set, without restarting the pointer to the current subject between epochs.

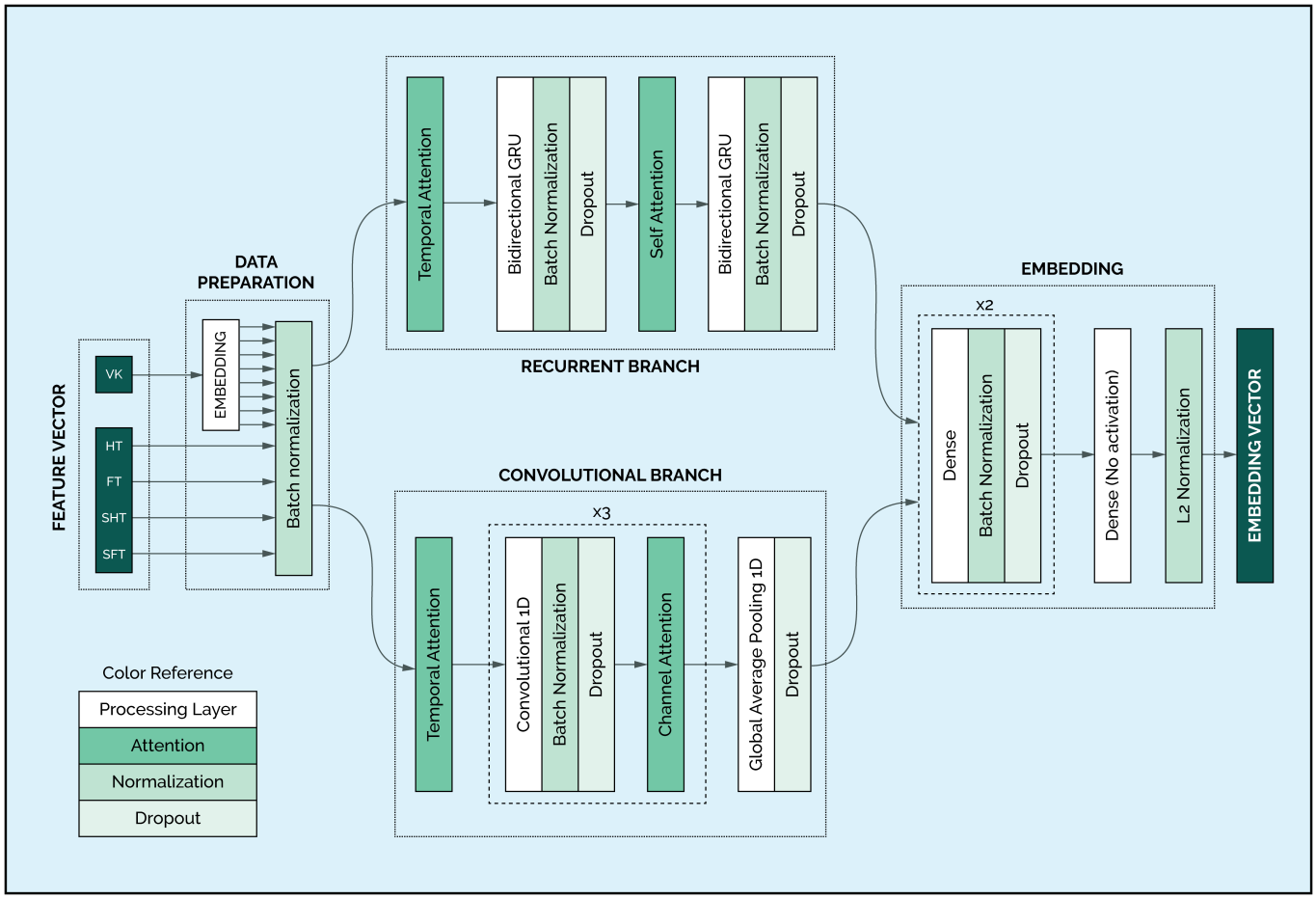


Fig. 3: Type2Branch: proposed dual-branch (recurrent and convolutional) embedding model for distance metric learning.

TABLE II: Type2Branch: Main hyperparameters.

Hyperparameter	Layers	Value
Number of units	GRU/dense	512
Number of units	embedding	64
Filters	conv1D	256, 512, 1024
Kernel size	conv1D	6
Activation	GRU layers	tanh
Activation	conv1D/dense	ReLU
Dropout rate		0.5

- m *nearest subjects*, determined as those whose centers of their embeddings are the m nearest to the center of the embeddings of the sequential subject. The embedding centers are calculated at the start of each epoch.
- $K-1-m$ *random subjects* chosen from all those available in the development set, with reposition between batches but making sure no batch includes repeated subjects.

D. Model Architecture

The proposed Type2Branch architecture uses two branches, one recurrent and one convolutional, to learn how to embed the input samples into a lower dimensional space. Fig. 3 shows the details of the proposed architecture. Sample similarity is finally measured by using the Euclidean distance between their respective embeddings, as traditionally done in DML.

The choice of the architecture is motivated by the observation that keystroke timings result from a combination of two factors: a partially conscious decision process involving *what* to type and an entirely unconscious motor process pertaining to *how* to type [19]. The convolutional branch is expected to excel at identifying common, short sequences, while the recurrent branch is expected to capture the subjects's time-dependent decision process. The rationale for combining both is described in Section VI-B, where one of the pilot experiments shows that a dual-branch model outperforms a purely recurrent or purely convolutional, single-branch model with an equivalent number of trainable parameters.

The proposed model is composed of four main modules: a small data preparation module whose only purpose is to embed the key code into a small dimensional space and normalize the input, the recurrent and convolutional branches that give the name to the proposed Type2Branch model, and a final embedding module. The main hyperparameters of the model are listed in Table II.

The recurrent branch comprises two bidirectional GRU layers (512 units), while the convolutional branch features three blocks of 1D convolution, each with an increasing number of filters (256, 512, and 1024, with kernel size equal to 6) and utilizes global average pooling. Temporal attention serves as the first layer of both branches. Scaled dot-product

self attention is applied between the recurrent layers, whereas channel attention follows each convolutional layer.

At the embedding module, the outputs of both branches are concatenated, and the final embedding vector is produced by three dense layers. Batch normalization and dropout are applied after each processing layer in all modules.

E. Set2set Loss Function - Motivation

The objective of the Set2set loss function is to minimize the EER of a keystroke dynamics verification system, operating under the conditions of one-shot evaluation. Moreover, it is assumed that a uniform global detection threshold is used across all subjects. There is no loss of generality in this assumption, as will be shown later.

In this scenario, the system makes a verification decision with a single reference sample per subject. In other words, it outputs a measure of similarity between a pair of samples: one that certainly belongs to the legitimate subject, and another from the subject under scrutiny. If the similarity value is below the global threshold, the sample is flagged as legitimate, or otherwise as an impostor.

Our initial point is the SetMargin loss proposed by Morales *et al.* in [20], which itself extends the well-known triplet loss function proposed by Schroff *et al.* [55]. Triplet loss aims to minimize the distance between samples of the same class while simultaneously enforcing a separation between samples of different classes by, at least, a given margin. SetMargin loss aims to capture better intra-class dependencies while enlarging the inter-class differences in the feature space, particularly along their boundaries where most classification errors occur, by adding the context of the set to the learning process.

We further extend the *SetMargin* loss by letting it compare, simultaneously, arbitrarily sized sets of sets instead of just pairs of sets, while also including an additive penalty term to encourage the model to embed all classes within hyperspheres with similar average radii. The extension to arbitrarily sized sets of sets, coupled with an adequate learning curriculum, allows the loss function to capture both the global and local structure of the embedding space more effectively. Incorporating an additive penalty to address variations in the average radius among different classes improves the EER when using a uniform global detection threshold, driving it closer to the average EER achieved with optimal thresholds per subject.

F. Set2set Loss Function - Formulation

Let $d(\mathbf{x}_i^m, \mathbf{x}_j^n)$ be the Euclidean distance between the embedding vectors corresponding to the i–th sample of the m–th class and the j–th sample of the n–th class, and define

$$L(\mathbf{x}_i^m, \mathbf{x}_j^m, \mathbf{x}_k^n) = d^2(\mathbf{x}_i^m, \mathbf{x}_j^m) - d^2(\mathbf{x}_i^m, \mathbf{x}_k^n) + \alpha \qquad (12)$$

The *Triplet Loss* function [55] with anchor $\mathbf{x}_i^m$, positive $\mathbf{x}_j^m$, and negative $\mathbf{x}_k^n$ is then

$$\mathcal{L}_{TL}(\mathbf{x}_i^m, \mathbf{x}_j^m, \mathbf{x}_k^n) = \max\{0, L(\mathbf{x}_i^m, \mathbf{x}_j^m, \mathbf{x}_k^n)\} \qquad (13)$$

The roles of the classes m and n can be made symmetric by defining

$$\mathcal{L}_{STL}(m, n, i, j, k) = \mathcal{L}_{TL}(\mathbf{x}_i^m, \mathbf{x}_j^m, \mathbf{x}_k^n) + \mathcal{L}_{TL}(\mathbf{x}_i^n, \mathbf{x}_j^n, \mathbf{x}_k^m) \qquad (14)$$

and in terms of $\mathcal{L}_{STL}$, the *SetMargin Loss* function [55] for the m–th and n–th classes is given by

$$\mathcal{L}_{SM}(m, n) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{N} \mathcal{L}_{STL}(m, n, i, j, k) \qquad (15)$$

Generalizing the above to the case when different classes have a varying number of samples is trivial; however, the assumption that all classes have N samples simplifies the exposition and the implementation of the proposed loss function, while allowing for significant optimizations in computing time when the latter is vectorized. Now, using the previous $\mathcal{L}_{SM}$ and the radius penalty from equation (3), we can define the proposed Set2set loss, which has the form

$$\mathcal{L}_{S2S} = \beta\, \mathcal{L}_{RP} + \sum_{m=1}^{K} \sum_{n=m+1}^{K} \mathcal{L}_{SM}(m, n) \qquad (16)$$

where $\mathcal{L}_{RP}$ is the radius penalty defined in equation (3). The number of sets K is a parameter to be optimized; in general, increasing K improves the performance until a certain limit given by the capacity of the model.

In a practical implementation, the constant β must be small enough so it does not interfere with the $\mathcal{L}_{SM}$ terms when encouraging the model to learn the structure of the embedding space, but large enough to slowly but surely enforce the normalization of the mean radii within the classes.

IV. Description of Databases

The experimental framework being proposed relies on the two most comprehensive and extensive public databases of free-text keystroke dynamics available to date. The raw data within the Aalto mobile keystroke database consists of Unix timestamps capturing key press and release actions, each timestamp having a 1 ms-resolution and being associated with the specific ASCII code of the key pressed. The data collection occurred through a mobile web application in a completely unsupervised manner. Participants were instructed to read, memorize, and type English sentences presented on their smartphones. These sentences were randomly selected from a pool of 1,525 sentences sourced from the Enron mobile mail [56] and the Gigaword Newswire corpora [57]. Hence, the text format employed is transcript-text, meaning the content was not formulated by the participants themselves, and the sentences contained at least 70 characters. Additionally, volunteers were instructed to type as quickly and accurately as possible. Notably, the volunteers were scattered in 163 countries, with English native speakers constituting around 68% of the participant pool.

Both databases are available for download in the form provided within the KVC[9]. They are organized into four datasets, with some subjects being excluded due to insufficient

[9]https://codalab.lisn.upsaclay.fr/competitions/14063

TABLE III: Demographic distributions of the provided datasets. The rows represent different age groups, while the columns represent genders. The evaluation sets are balanced with respect to gender.

Task 1: Desktop Dataset

Development Set		
	Male	**Female**
10 - 13	4,336	5,420
14 - 17	10,993	8,336
18 - 26	25,752	24,315
27 - 35	9,607	12,281
36 - 44	2,143	5,331
45 - 79	1,182	5,424
Total Labelled: 115,120, Total unlabeled: 0		

Evaluation Set		
	Male	**Female**
10 - 13	1,085	1,085
14 - 17	1,861	1,861
18 - 26	1,861	1,861
27 - 35	1,861	1,861
36 - 44	536	536
45 - 79	296	296
Total Labelled: 15,000, Total unlabeled: 0		

Task 2: Mobile Dataset

Development Set		
	Male	**Female**
10 - 13	622	800
14 - 17	1,537	1,516
18 - 26	4,359	8,999
27 - 35	1,343	4,002
36 - 44	382	1,333
45 - 79	200	739
Total Labelled: 25,832, Total unlabeled: 14,807[10]		

Evaluation Set		
	Male	**Female**
10 - 13	254	254
14 - 17	618	618
18 - 26	843	843
27 - 35	547	547
36 - 44	156	156
45 - 79	82	82
Total Labelled: 5,000, Total unlabeled: 0		

acquisition sessions per subject (fewer than 15 samples per subject):

- Desktop Dataset:
  - Development set: 115,120 subjects, with an average sample length of 48.65 (σ = 18.50) characters typed.
  - Evaluation set: 15,000 subjects, with an average sample length of 48.77 (σ = 18.64) characters typed.
- Mobile Dataset:
  - Development set: 40,639 subjects, with an average sample length of 48.59 (σ = 21.84) characters typed.
  - Evaluation set: 5,000 subjects, with an average sample length of 47.98 (σ = 20.93) characters typed.

In each dataset, all subjects have undergone at least 15 acquisition sessions. The experimental framework proposed adopts an open-set learning protocol, meaning the subjects in the development and evaluation sets are distinct (see Sec. V). While a validation set is not explicitly provided, it can be derived from the development set using various training approaches.

Thanks to the demographic (age, gender) labels present in the original database, the subjects in the provided datasets have been arranged to enable a study of the demographic differentials in the scores for purposes such as privacy quantification and fairness [17]. Table III shows the demographic distribution of the datasets provided in the KVC. The subjects have been divided into six age groups (10 - 13, 14 - 17, 18 - 26, 27 - 35, 36 - 44, 45 - 79). The evaluation sets are balanced with respect to gender. The gender and age labels are available for download alongside the development set files.

[10]Although unlabeled, we opted to include these subjects to maximize the size of the provided dataset.

V. Experimental Protocol

A. Model Training

The batch structure has been described in Section III-C. Values of $N = 15$ samples per subject and $K = 40$ subjects per batch are used. The value of $K = 40$ is chosen as the largest that allows the optimized implementation to fit in the GPU memory. The pilot experiments (see Section VI-B) show that increasing K improves the classification accuracy of the trained model.

Each epoch consists of 20,000 steps for the desktop scenario and 7,000 steps for the mobile scenario. Validation loss is calculated at the end of each epoch. An early stopping strategy leveraging it, with a minimum delta of 10^{-4} and a patience of 12 epochs, is used to determine the optimal duration of the training, which lasted 18 epochs. Only the best model, as measured by the validation loss, is saved.

The standard Adam optimizer is used for training the model. The learning rate is set to 10^{-4}, while the rest of the parameters are left at default values, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. No scheduling is used; the learning rate is kept fixed throughout the entire training process.

This value of β in equation (16) is set to 0.05 in the implementation of the loss function.

B. Tools and Frameworks

The model is trained with Keras 2.11.0 and Tensorflow 2.12, running in Python 3.10.10 and using an NVIDIA A100 40GB GPU. The Set2set loss is implemented as a tensorflow function and optimized for computing speed, given that a naïve implementation of the deeply nested loop implicit in equation (16) is prohibitively slow even for small K.

The synthetic timing features meant to reflect how the typing style of a subject differs from that of the general population were synthesized with the tool [23], which binary and source code are publicly available.

C. Evaluation Description

The KVC standard experimental protocol is adopted in the experiments of the present study. The scores generated by the biometric systems are submitted to CodaLab to obtain all the metrics reported in Sec. VI.

The approach described next is valid for both desktop and mobile cases in a verification scheme. The comparison list provided in the KVC competition specifies the comparisons to perform between samples.

The total count of 1 vs. 1 sample-level comparisons is as follows:

- Task 1 (Desktop): 2,250,000 comparisons, involving 15,000 subjects not present in the development set.
- Task 2 (Mobile): 750,000 comparisons, involving 5,000 subjects not present in the development set.

For each subject in the evaluation sets, we use 5 samples for enrollment and 10 samples for verification. By considering all possible genuine pairwise comparisons, we obtain 50 comparison scores. These scores are then averaged over the 5 enrollment samples, resulting in 10 genuine scores per subject.

TABLE IV: Comparison of the proposed Type2Branch with the state of the art on the evaluation dataset of KVC-ongoing [17].

Global Distributions

Experiment	EER (%)	FNMR@0.1% FMR (%)	FNMR@1% FMR (%)	FNMR@10% FMR (%)	AUC (%)	Accuracy (%)
Desktop						
TypeNet [25]	6.76	77.4	39.57	3.45	98.08	93.24
TypeFormer [47]	12.75	94.75	73.53	18.19	94.32	87.25
Type2Branch	**3.33**	**44.17**	**11.96**	**0.51**	**99.48**	**96.68**
Mobile						
TypeNet [25]	13.95	92.76	70.05	22.22	93.8	86.05
TypeFormer [47]	9.45	94.77	67.67	8.53	96.22	90.55
Type2Branch	**3.61**	**63.62**	**17.44**	**0.60**	**99.28**	**96.39**

Mean Per-subject Distributions

Experiment	EER (%)	AUC (%)	Accuracy (%)	Rank-1 (%)
Desktop				
TypeNet [25]	2.71	99.26	95.31	89.81
TypeFormer [47]	7.76	96.51	91.13	65.52
Type2Branch	**0.77**	**99.87**	**96.43**	**98.04**
Mobile				
TypeNet [25]	7.99	96.43	90.89	68.5
TypeFormer [47]	5.25	97.89	93.28	75.92
Type2Branch	**1.03**	**99.76**	**96.24**	**96.11**

Similarly, 20 impostor scores are generated per subject. The impostor samples are divided into two groups: 10 similar impostor scores, where verification samples are randomly selected from subjects of the same demographic group (same gender and age), and 10 dissimilar impostor scores, where verification samples are all randomly selected from subjects of different gender and age intervals.

Based on the described evaluation design, two approaches are followed to evaluate the system:

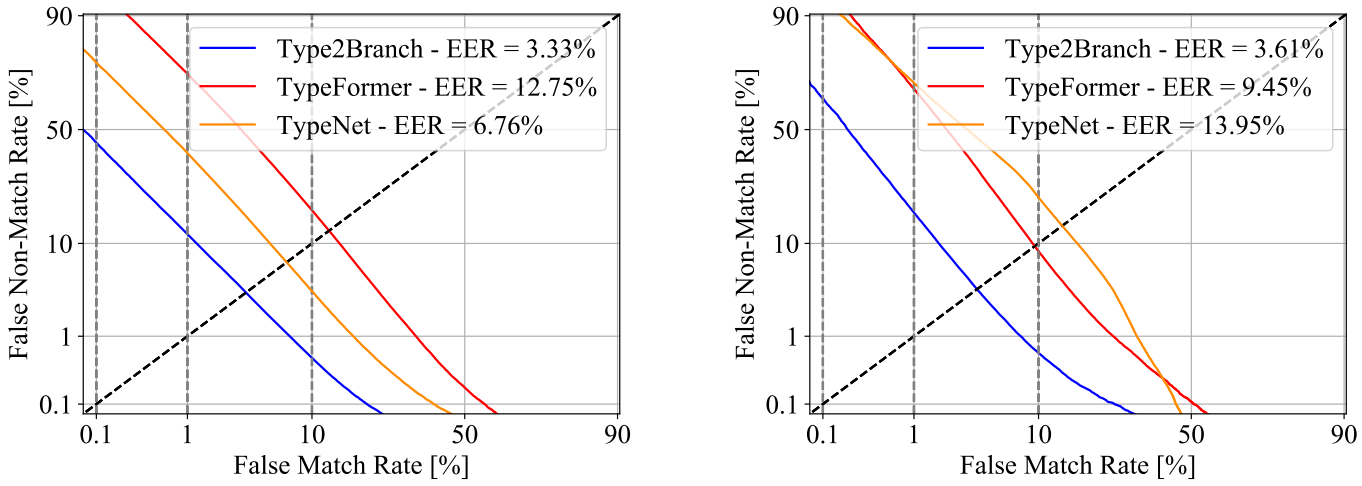
- Global distributions: all scores are divided into two groups, genuine and impostor scores, regardless of the subjects they belong to. This approach entails using a fixed, pre-determined threshold, simplifying the deployment of the biometric system. Performance assessment involves setting a single threshold for all comparisons to reach a decision.
- Mean per-subject distributions: the optimal threshold is computed at the subject-level, considering the 30 verification scores as described above. This approach offers more flexibility, allowing the system to adapt to subject-specific distributions. In real-life scenarios, this would involve processing the subject's enrollment samples to establish a threshold. The process includes acquiring various enrollment samples to derive a genuine subject-specific score distribution and considering a pool of samples from different subjects to derive an impostor subject-specific score distribution. A subject-specific threshold is then computed based on these distributions. Importantly, this doesn't necessitate re-training or fine-tuning the biometric system using subject-specific data. Metrics computed per-subject are averaged across all subjects in the evaluation set to obtain the displayed values. Generally, the system's verification performance benefits from employing a different threshold per subject.

VI. Experimental Results

A. Biometric recognition

The verification performance of the proposed Type2Branch is reported in Table IV. For all metrics, the reported values are computed on the KVC evaluation set. Currently, Type2Branch achieves the highest verification performance among all systems proposed in the ongoing competition [24]. To provide a comparison with state-of-the-art KD systems, Table IV also features the results achieved by TypeNet [25] and TypeFormer [47] over the same experimental setup.

As can be seen, Type2Branch improves previous verification records in all cases, by a significant margin. Considering global distributions, the EER obtained by Type2Branch (3.33%) is halved in comparison with TypeNet (6.75%) in the desktop case, and it is reduced to almost one third of the EER scored by TypeFormer in the mobile case (3.61% vs 9.45%). Moving along the columns of Table IV, the gap dramatically widens in the case of different operational points, *e.g.*, False Non-Match Rate (FNMR) at 1%, 10% of False Match Rate (FMR). The described trends are also consistent when analyzing the mean per-subject distributions. For instance, as displayed in the bottom half of Table IV, Type2Branch achieves a 0.77% EER *vs* 2.71% EER (TypeNet) for desktop, and 1.03% EER *vs* 5.25% EER (TypeFormer) for mobile. Moreover, in contrast with all metrics which are related to the task of verification, the rank-1 metric reported is related to the task of identification, and it represents the percentage of times in which from a subject-specific pool of 21 samples (20 impostor samples and 1 genuine sample), the genuine one achieves the highest score [17]. Consequently, given this formulation, it can only be applied to subject-specific distributions. Once again, Type2Branch significantly outperforms previous approaches, showing that the Set2set loss proposed is able to map the embedding space in a much finer way with respect to the triplet loss [55] or the original



(a) Desktop task.

(b) Mobile task.

Fig. 4: DET curves including the results of all the biometric verification systems analyzed. The grey dashed lines indicate the operational points 0.1% FMR, 1% FMR, and 10% FMR, whereas the black dashed line indicate the points where the FMR = FNMR, corresponding to the EER.

SetMargin loss [20], leading to unprecedented results in the identification task as well.

Fig. 4 graphically depicts such trends in the form of Detection Error Trade-off (DET) curves. In particular, from left to right, Fig. 4(a) and Fig. 4(b) respectively show the desktop and mobile cases. The grey dashed lines indicate the operational points 0.1% FMR, 1% FMR and 10% FMR whereas the black dashed line indicate the points where the FMR = FNMR, corresponding to the EER point. Noteworthily, in the field of behavioral biometrics a threshold corresponding to FMR = 1% represents a rather safe system from the perspective of limiting the intrusions, and this generally leads to a significant degradation of the FNMR, as proved in the case of both TypeFormer and TypeNet. Nevertheless, Type2Branch is able to limit the performance drop to 11.86% and 17.44% FNMR. Such results are very promising considering that Type2Branch is not developed nor optimized for FNMR/FMR metric.

B. Pilot Experiments

The design of the final Type2Branch model, whose results have been described in the previous section, was obtained after several pilot experiments, using all users and samples of the development dataset. These experiments proved the efficacy of various architectural and training choices in smaller, baseline models. We think these intermediate results are interesting enough to be reproduced here.

Early in the development cycle, we observed a noticeable improvement on the model's performance when the number of sets simultaneously considered by the Set2set loss function was increased. As Figure 5 shows in blue, even a small baseline model trained with 1,000 users experiences noticeable reductions in the global EER as the value of K is increased from 2 to 10. Although the performance declines for $K > 10$ in this case, we conjectured that a larger model trained with a larger dataset would benefit from increasing K further. This

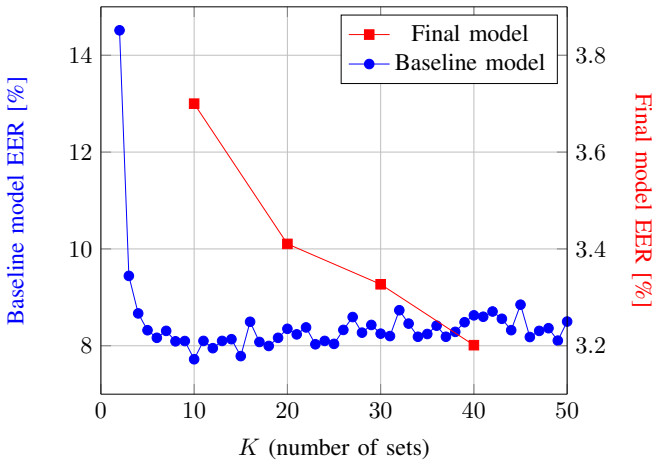


Fig. 5: Effect on the EER with a global threshold when increasing K (number of sets) in the loss function. We compare a baseline model trained with 1K users with the final model. Given enough training data, a large K improves the performance by providing a more comprehensive purview of the embedding space.

is in fact proved in Figure 5, in red, for the case of the final model, trained with the entirety development dataset. Given additional GPU memory, allowing for even larger values of K, it is plausible that the performance can be improved further.

Figure 6 shows the mean per-user EER for a baseline model, trained with 2K users and employing an architecture analogous to the final model but with fewer parameters. In this experiment, the performance of individual branches has been explored for a varying number of enrollment samples $G = 1, 2, 5, 7, 10$. Note that the parameter count for the single-branch models was calibrated to match that of the dual-branch model. This pilot experiment shows that the dual-branch model outperforms single-branch configurations by $\approx 25\%$.

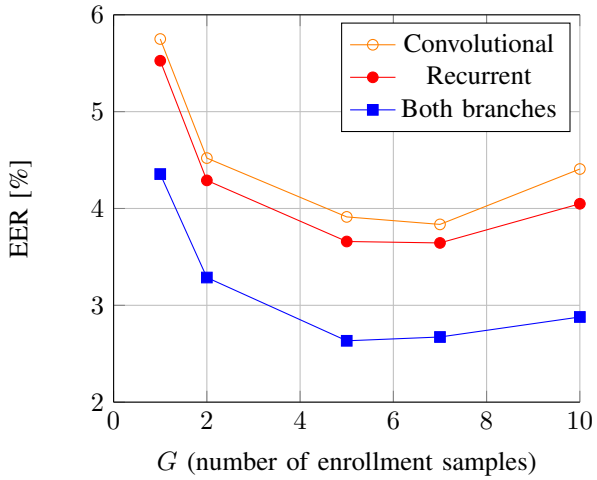


Fig. 6: Average user EER of single- and dual-branch models with 2K training users. For the same number of parameters, a dual-branch model performs better than each branch separately.

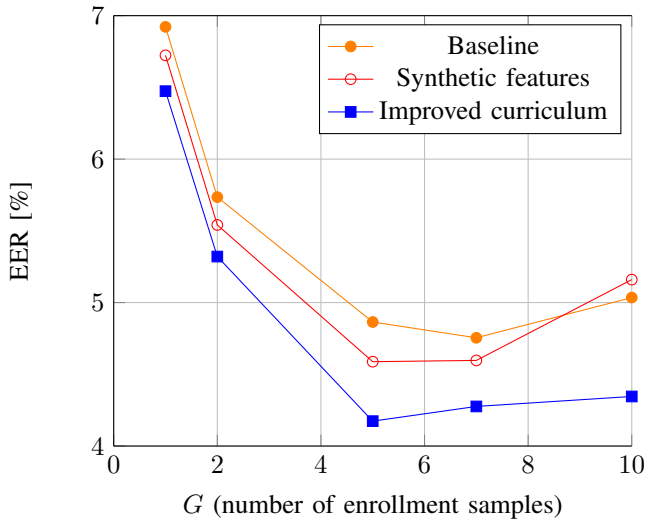


Fig. 7: Average user EER of a baseline model trained with 1K users, compared with improvements on learning curriculum and synthetic features.

Finally, Figure 7 shows the changes in mean per-user EER when adding synthetic features and modifications to the training curriculum of a baseline dual-branch model trained with 1K users. While these enhancements produce more modest performance gains when compared to those in previous experiments, they collectively achieve a relative reduction in EER of $\approx$ 10-15% with minimal overhead.

C. Limitations

The proposed model has been trained on a balanced subset of the Aalto datasets, which consist of typing samples from transcribed text. Unfortunately, there are no publicly available datasets comprising free-text samples that are sufficiently large for training and evaluating a model of this size and complexity.

Furthermore, the evaluation of the proposed model was conducted using an attack model that assumes zero-effort impostors, meaning that the impostor samples do not exhibit a deliberate attempt to emulate the typing style of the legitimate users. While it is standard practice to assess behavioral biometric systems under this assumption, conducting evaluations under more advanced attack models could potentially reveal decreased performance.

VII. Conclusions and Future Work

In the present study, we have proposed and evaluated the performance of Type2Branch, a dual-branch (recurrent and convolutional) distance metric learning model for the task of verifying user identities based on their keystroke dynamics. Following the recent trend initiated by TypeNet and continued by TypeFormer, our approach leverages large datasets to train a deep learning model that can scale to hundreds of thousands of users with little performance loss.

To the best of our knowledge, the proposed Type2Branch outperforms the state of the art with a mean per-user EER of 0.77% in the desktop scenario and 1.03% in the mobile scenario when using ten enrollment samples. Under a harder evaluation criteria, when only a single enrollment sample per user is available and a uniform global threshold is used, the proposed model still manages to achieve an EER of 3.33% in the desktop scenario and 3.61% in the mobile scenario. These results have been validated with a sound experimental protocol and using publicly available training and evaluation datasets during the KVC-onGoing competition [24].

The pilot experiments showed that a dual-branch architecture outperforms single-branch architectures for a given number of parameters; also, that several gradual improvements like the proposed synthetic features and the improved training curriculum provide cumulative gains. Nevertheless, the most noticeable improvement was provided by the Set2set loss that, by extending SetMargin loss to larger numbers of sets, allows the model to optimize the embedding space globally and results in noticeable performance gains.

Future work will be oriented towards exploring architectures with more than two branches, improving the branches of the proposed model, and evaluating the generalizability and utility of the Set2set loss function for other biometric verification and general classification tasks. We plan on making the implementation of the Set2set loss function available soon.

Acknowledgments

This research work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 860315, projects INTER-ACTION (PID2021-126521OB-I00 MICINN/FEDER) and Cátedra ENIA UAM-VERIDAS en IA Responsable (NextGenerationEU PRTR TSI-100927-2023-2), and by Comunidad de Madrid (ELLIS Unit Madrid). The authors would like to thank Ms. Susan Essex for proofreading and language editing this manuscript, and Brian Callipari for providing the illustrations.

REFERENCES

[1] P. Delgado-Santos, R. Tolosana, R. Guest, R. Vera-Rodriguez, and J. Fierrez, "M-GaitFormer: Mobile biometric gait verification using Transformers," *Engineering Applications of Artificial Intelligence*, vol. 125, p. 106682, 2023.

[2] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, "BioTouchPass2: Touchscreen password biometrics using time-aligned recurrent neural networks," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2616–2628, 2020.

[3] J. Fierrez, A. Pozo, M. Martinez-Diaz, J. Galbally, and A. Morales, "Benchmarking touchscreen biometrics for mobile authentication," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2720–2733, 2018.

[4] R. Tolosana, R. Vera-Rodriguez, and *et al.*, "SVC-onGoing: Signature verification competition," *Pattern Recognition*, vol. 127, 2022.

[5] G. Stragapede, R. Vera-Rodriguez, R. Tolosana, and A. Morales, "BehavePassDB: public database for mobile behavioral biometrics and benchmark evaluation," *Pattern Recognition*, vol. 134, p. 109089, 2023.

[6] A. Morales, J. Fierrez, R. Tolosana, J. Ortega-Garcia, J. Galbally, M. Gomez-Barrero, A. Anjos, and S. Marcel, "Keystroke biometrics ongoing competition," *IEEE Access*, vol. 4, pp. 7736–7746, 2016.

[7] R. L. Mandryk, J. Frommel, N. Goyal, G. Freeman, C. Lampe, S. Vieweg, and D. Y. Wohn, "Combating Toxicity, Harassment, and Abuse in Online Social Spaces: A Workshop at CHI 2023," in *Proc. Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–7.

[8] P. R. Borj, K. Raja, and P. Bours, "Online grooming detection: A comprehensive survey of child exploitation in chat logs," *Knowledge-Based Systems*, vol. 259, p. 110039, 2023.

[9] A. Morales, A. Acien, J. Fierrez, J. V. Monaco, R. Tolosana, R. Vera, and J. Ortega-Garcia, "Keystroke biometrics in response to fake news propagation in a global pandemic," in *Proc. IEEE Annual Computers, Software, and Applications Conference (COMPSAC)*, 2020, pp. 1604–1609.

[10] O. Benjakob, "Netanyahu vs. Israeli Security Chiefs: Wikipedia Is New Front in Gaza War Blame Game," *Haaretz*, 2023-11-17.

[11] P. Melzi, C. Rathgeb, R. Tolosana, R. Vera-Rodriguez, A. Morales, D. Lawatsch, F. Domin, and M. Schaubert, "Synthetic Data for the Mitigation of Demographic Biases in Face Recognition," in *Proc. IEEE International Joint Conference on Biometrics (IJCB)*, 2023, pp. 1–9.

[12] I. DeAndres-Tame, R. Tolosana, R. Vera-Rodriguez, A. Morales, J. Fierrez, and J. Ortega-Garcia, "How Good is ChatGPT at Face Biometrics? A First Look into Recognition, Soft Biometrics, and Explainability," *IEEE Access*, 2024.

[13] A. A. N. Buker, G. Roffo, and A. Vinciarelli, "Type Like a Man! Inferring Gender from Keystroke Dynamics in Live-Chats," *IEEE Intelligent Systems*, vol. 34, no. 6, pp. 53–59, 2019.

[14] A. Pentel, "Predicting user age by keystroke dynamics," in *Proc. Artificial Intelligence and Algorithms in Intelligent Systems*, 2019, pp. 336–343.

[15] A. N. H. Nahin, J. M. Alam, H. Mahmud, and K. Hasan, "Identifying emotion by keystroke dynamics and text pattern analysis," *Behaviour & Information Technology*, vol. 33, no. 9, pp. 987–996, 2014.

[16] I. Tsimperidis, D. Grunova, S. Roy, and L. Moussiades, "Keystroke dynamics as a language profiling tool: Identifying mother tongue of unknown internet users," *Telecom*, vol. 4, no. 3, pp. 369–377, 2023.

[17] G. Stragapede, R. Vera-Rodriguez, R. Tolosana, A. Morales, N. Damer, J. Fierrez, and J. Ortega-Garcia, "Keystroke Verification Challenge (KVC): Biometric and Fairness Benchmark Evaluation," *IEEE Access*, 2023.

[18] K. S. Killourhy and R. A. Maxion, "Free vs. transcribed text for keystroke-dynamics evaluations," in *Proceedings of the 2012 Workshop on Learning from Authoritative Security Experiment Results*, 2012, pp. 1–8.

[19] N. González, E. P. Calot, J. S. Ierache, and W. Hasperué, "On the shape of timings distributions in free-text keystroke dynamics profiles," *Heliyon*, vol. 7, no. 11, p. e08413, 2021.

[20] A. Morales, J. Fierrez, A. Acien, R. Tolosana, and I. Serna, "SetMargin loss applied to deep keystroke biometrics with circle packing interpretation," *Pattern Recognition*, vol. 122, p. 108283, 2022.

[21] R. Tolosana, P. Delgado-Santos, A. Perez-Uribe, R. Vera-Rodriguez, J. Fierrez, and A. Morales, "DeepWriteSYN: On-line handwriting synthesis via deep short-term representations," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 600–608.

[22] P. Melzi, R. Tolosana, R. Vera-Rodriguez, and *et al.*, "FRCSyn-onGoing: Benchmarking and comprehensive evaluation of real and synthetic data to improve face recognition systems," *Information Fusion*, vol. 107, p. 102322, 2024.

[23] N. González, "KSDSLD — A tool for keystroke dynamics synthesis & liveness detection," *Software Impacts*, vol. 15, p. 100454, 2023.

[24] G. Stragapede, R. Vera-Rodriguez, R. Tolosana, A. Morales, I. DeAndres-Tame, N. Damer, J. Fierrez, J.-O. Garcia, N. Gonzalez, A. Shadrikov *et al.*, "IEEE BigData 2023 Keystroke Verification Challenge (KVC)," in *Proc. 2023 IEEE International Conference on Big Data (BigData)*. IEEE, 2023, pp. 6092–6100.

[25] A. Acien, A. Morales, J. V. Monaco, R. Vera-Rodriguez, and J. Fierrez, "TypeNet: Deep Learning Keystroke Biometrics," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 1, pp. 57–70, 2021.

[26] E. Maiorana, H. Kalita, and P. Campisi, "Mobile keystroke dynamics for biometric recognition: An overview," *IET Biometrics*, vol. 10, no. 1, pp. 1–23, 2021.

[27] S. Roy, J. Pradhan, A. Kumar, D. R. D. Adhikary, U. Roy, D. Sinha, and R. K. Pal, "A systematic literature review on latest keystroke dynamics based models," *IEEE Access*, 2022.

[28] Y. Deng and Y. Zhong, "Keystroke Dynamics User Authentication Based on Gaussian Mixture Model and Deep Belief Nets," *International Scholarly Research Notices*, vol. 2013, 2013.

[29] K. S. Killourhy and R. A. Maxion, "Comparing anomaly-detection algorithms for keystroke dynamics," in *Proc. Int. Conf. on Dependable Systems & Networks*, 2009.

[30] A. A. Ahmed and I. Traore, "Biometric Recognition Based on Free-Text Keystroke dynamics," *IEEE Transactions on Cybernetics*, vol. 44, no. 4, pp. 458–472, 2013.

[31] H. Çeker and S. Upadhyaya, "Sensitivity Analysis in Keystroke Dynamics using Convolutional Neural Networks," in *Proc. Workshop on Information Forensics and Security*, 2017.

[32] O. Alpar, "Keystroke recognition in user authentication using ANN based RGB histogram technique," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 213–217, 2014.

[33] I. Stylios, A. Skalkos, S. Kokolakis, and M. Karyda, "BioPrivacy: Development of a Keystroke Dynamics Continuous Authentication System," in *Proc. Computer Security. ESORICS 2021 Int. Workshops*, 2022.

[34] X. Lu, S. Zhang, P. Hui, and P. Lio, "Continuous Authentication by Free-Text Keystroke Based on CNN and RNN," *Computers & Security*, vol. 96, p. 101861, 2020.

[35] Y. Sun, H. Ceker, and S. Upadhyaya, "Shared keystroke dataset for continuous authentication," in *Proc. IEEE Int. Workshop on Information Forensics and Security*, 2016.

[36] E.-S. M. El-Kenawy, S. Mirjalili, A. A. Abdelhamid, A. Ibrahim, N. Khodadadi, and M. M. Eid, "Meta-Heuristic Optimization and Keystroke Dynamics for Authentication of Smartphone Users," *Mathematics*, vol. 10, no. 16, 2022.

[37] J. Li, H.-C. Chang, and M. Stamp, "Free-Text Keystroke Dynamics for User Authentication," *Artificial Intelligence for Cybersecurity*, pp. 357–380, 2022.

[38] V. Dhakal, A. M. Feit, P. O. Kristensson, and A. Oulasvirta, "Observations on Typing from 136 Million Keystrokes," in *Proc. CHI Conf. on Human Factors in Computing Systems*, 2018.

[39] K. Palin, A. M. Feit, S. Kim, P. O. Kristensson, and A. Oulasvirta, "How Do People Type on Mobile Devices? Observations from a Study with 37,000 Volunteers," in *Proc. Int. Conf. on Human-Computer Interaction with Mobile*, 2019.

[40] R. Giot, M. El-Abed, and C. Rosenberger, "Greyc keystroke: a benchmark for keystroke dynamics biometric systems," in *Proc. Int. Conf. on Biometrics: Theory, Applications, and Systems*, 2009.

[41] J. Fierrez, J. Galbally, J. Ortega-Garcia, M. R. Freire, F. Alonso-Fernandez, D. Ramos, D. T. Toledano, J. Gonzalez-Rodriguez, J. A. Siguenza, J. Garrido-Salas *et al.*, "BiosecurID: a multimodal biometric database," *Pattern Analysis and Applications*, vol. 13, pp. 235–246, 2010.

[42] M. El-Abed, M. Dafer, and R. El Khayat, "RHU Keystroke: A mobile-based benchmark for keystroke dynamics systems," in *Proc. Int. Carnahan Conf. on Security Technology*, 2014, pp. 1–4.

[43] E. Vural, J. Huang, D. Hou, and S. Schuckers, "Shared research dataset to support development of keystroke authentication," in *Proc. Int. Joint Conf. on Biometrics*, 2014.

[44] C. Murphy, J. Huang, D. Hou, and S. Schuckers, "Shared dataset on natural human-computer interaction to support continuous authentication research," in *Proc. Int. Joint Conf. on Biometrics*, 2017.

[45] A. Acien, A. Morales, J. Fierrez, R. Vera-Rodriguez, and O. Delgado-Mohatar, "BeCAPTCHA: Behavioral bot detection using touchscreen and mobile sensors benchmarked on HuMIdb," *Engineering Applications of Artificial Intelligence*, vol. 98, p. 104058, 2021.

[46] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons," *Advances in neural information processing systems*, vol. 16, 2003.

[47] G. Stragapede, P. Delgado-Santos, R. Tolosana, R. Vera-Rodriguez, R. Guest, and A. Morales, "TypeFormer: Transformers for mobile keystroke biometrics," *arXiv:2212.13075*, 2023.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Proc. Advances in Neural Information Processing Systems*, 2017.

[49] T. Neacsu, T. Poncu, S. Ruseti, and M. Dascalu, "DoubleStrokeNet: Bigram-Level Keystroke Authentication," *Electronics*, vol. 12, no. 20, p. 4309, 2023.

[50] D. Stefan, X. Shu, and D. D. Yao, "Robustness of keystroke-dynamics based biometrics against synthetic forgeries," *computers & security*, vol. 31, no. 1, pp. 109–121, 2012.

[51] J. V. Monaco, M. L. Ali, and C. C. Tappert, "Spoofing key-press latencies with a generative keystroke dynamics model," in *Proc. IEEE Int. Conf. on Biometrics Theory, Applications and Systems (BTAS)*, 2015, pp. 1–8.

[52] N. González, E. P. Calot, J. S. Ierache, and W. Hasperué, "Towards liveness detection in keystroke dynamics: Revealing synthetic forgeries," *Systems and Soft Computing*, vol. 4, p. 200037, 2022.

[53] D. DeAlcala, A. Morales, R. Tolosana, A. Acién, J. Fierrez, S. Hernández, M. A. Ferrer, and M. Diaz, "BeCAPTCHA-Type: Biometric Keystroke Data Generation for Improved Bot Detection," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023, pp. 1051–1060.

[54] N. Gonzalez and E. P. Calot, "Finite context modeling of keystroke dynamics in free text," in *Proc. International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2015, pp. 1–5.

[55] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE/CVF conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[56] K. Vertanen and P. O. Kristensson, "A Versatile Dataset for Text Entry Evaluations Based on Genuine Mobile Emails," in *Proc. Int. Conf. on Human Computer Interaction with Mobile Devices and Services*, 2011.

[57] D. Graff and C. Cieri, "English Gigaword LDC2003T05," *Philadelphia: Linguistic Data Consortium*, 2003.

Nahuel González received his PhD degree in Computing Science from Universidad Nacional de La Plata (UNLP), Argentina, in 2022. His PhD thesis was awarded the Raúl Gallard Prize, handed by the Network of Argentinian Universities of Computing Science (RedUNCI), the following year. He has been affiliated with the Laboratorio de Sistemas de Información Avanzados (LSIA) of the University of Buenos Aires (UBA) since 2013. His main research interests are behavioral biometrics and time series prediction/classification using deep learning. He is a member of the editorial board of Data in Brief, Elsevier, since 2024.

Giuseppe Stragapede received his MSc degree in electronic engineering from Politecnico di Bari, Italy, in 2019. After one year as a computer vision engineer in the industry, in 2020 he started his PhD with a Marie Curie Fellowship within the PriMa (Privacy Matters) EU project in the Biometrics and Data Pattern Analytics - BiDA Lab, at the Universidad Autonoma de Madrid, Spain. His research interests include biometrics (especially mobile biometrics), data protection, signal processing, and machine learning.

Ruben Vera-Rodriguez received his PhD degree in electrical and electronic engineering from Swansea University, U.K., in 2010. Since then, he has been affiliated with the Biometric Recognition Group, Universidad Autonoma de Madrid, Spain, where he is currently an Associate Professor since 2018. His research interests include signal and image processing, pattern recognition, HCI and biometrics, with emphasis on signature, face, gait verification, mobile biometrics and forensic applications of biometrics. He is actively involved in several national and European projects focused on biometrics. He has been awarded recently with a Medal from the Spanish Royal Academy of Engineering for his research contributions. He is member of ELLIS Society.

Ruben Tolosana received the M.Sc. degree in Telecommunication Engineering, and the Ph.D. degree in Computer and Telecommunication Engineering, from Universidad Autonoma de Madrid, in 2014 and 2019, respectively. In 2014, he joined the Biometrics and Data Pattern Analytics - BiDA Lab at the Universidad Autonoma de Madrid, where he is currently an Assistant Professor. He is a member of the ELLIS Society, the Technical Area Committee of EURASIP, and the Editorial Board of the IEEE Biometrics Council Newsletter. His research interests are mainly focused on signal and image processing, pattern recognition, and machine learning, particularly in the areas of DeepFakes, Human-Computer Interaction, Biometrics, and Health. He is the author of more than 90 scientific articles published in international journals and conferences. He has served as General Chair and Program Chair (AVSS 2022), and Area Chair (IJCB 2023, ICPR 2022) in top conferences. Dr. Tolosana has also received several awards such as the European Biometrics Industry Award (2018) from the European Association for Biometrics (EAB) and the Best Ph.D. Thesis Award in 2019-2022 from the Spanish Association for Pattern Recognition and Image Analysis (AERFAI).