Neural Computing and Applications
https://doi.org/10.1007/s00521-024-10140-2

ORIGINAL ARTICLE

TypeFormer: transformers for mobile keystroke biometrics

Giuseppe Stragapede[1] · Paula Delgado-Santos[2] · Ruben Tolosana[1] · Ruben Vera-Rodriguez[1] · Richard Guest[3] · Aythami Morales[1]

Received: 6 November 2023 / Accepted: 27 June 2024

Abstract
The broad usage of mobile devices nowadays, the sensitiveness of the information contained in them, and the shortcomings of current mobile user authentication methods are calling for novel, secure, and unobtrusive solutions to verify the users' identity. In this article, we propose TypeFormer, a novel transformer architecture to model free-text keystroke dynamics performed on mobile devices for the purpose of user authentication. The proposed model consists in temporal and channel modules enclosing two long short-term memory recurrent layers, Gaussian range encoding, a multi-head self-attention mechanism, and a block-recurrent transformer layer. Experimenting on one of the largest public databases to date, the Aalto mobile keystroke database, TypeFormer outperforms current state-of-the-art systems achieving equal error rate values of 3.25% using only five enrolment sessions of 50 keystrokes each. In such way, we contribute to reducing the traditional performance gap of the challenging mobile free-text scenario with respect to its desktop and fixed-text counterparts. To highlight the design rationale, an analysis of the experimental results of the different modules implemented in the development of TypeFormer is carried out. Additionally, we analyse the behaviour of the model with different experimental configurations such as the length of the keystroke sequences and the amount of enrolment sessions, showing margin for improvement.

Keywords Keystroke dynamics · Transformers · Biometrics · Mobile devices · HCI

1 Introduction

The rapid digitalisation of the society, together with the pervasiveness of mobile devices, is making room for unprecedented human–computer interaction (HCI) scenarios. Most people are now constantly connected to the internet through their mobile devices, accessing remotely their private data, and carrying out sensitive operations in sectors such as Banking, Financial Services and Insurance (BFSI), healthcare, e-commerce, and government, among many others [1]. This trend has increased the amount of cybercrimes observed [2], evidencing the need for novel and reliable security methods that fulfil context-specific constraints, such as: (i) continuous protection; (ii) user-friendliness; (iii) limited processing load, compatible with

✉ Giuseppe Stragapede
giuseppe.stragapede@estudiante.uam.es

Paula Delgado-Santos
paula.delgadodesantos@telefonica.com

Ruben Tolosana
ruben.tolosana@uam.es

Ruben Vera-Rodriguez
ruben.vera@uam.es

Richard Guest
r.m.guest@soton.ac.uk

Aythami Morales
aythami.morales@uam.es

[1] Biometrics and Data Pattern Analytics (BiDA) Lab, Universidad Autonoma de Madrid, 28049 Madrid, Spain

[2] Telefonica Research, Barcelona, Spain

[3] School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, United Kingdom

Published online: 30 July 2024

mobile environment specifications; and (iv) immunity to spoofing. To meet such requirements, recent studies have explored the feasibility of the user's behavioural[1] biometric traits as an authentication method to create an additional transparent security layer on top of traditional approaches [3, 4]. In fact, such traits can be constantly verified in a passive way [5, 6], i.e. without having the user to carry out any specific *entry-point* authentication task, such as placing their fingertip on the dedicated sensor, or typing a pass code, thus addressing (i) and (ii). Such methods are also convenient as mobile devices come equipped with several sensors that can be treated as sources of biometric modalities [7, 8]. Mobile behavioural biometric traits are also captured as low-dimensional time-domain signals, i.e. the acquisition and processing is fast (iii). Additionally, it has been argued that spoofing behavioural biometrics requires more advanced technical skills compared to their physiological counterparts (iv) [2]. Keystroke dynamics represents one of the most popular and high-performance authentication methods among mobile behavioural biometrics [9].

In the present work, we propose a novel transformer architecture, TypeFormer, for mobile keystrokes dynamics for the purpose of user authentication. Transformers are recent deep learning (DL) networks, originally characterised by an encoder–decoder architecture [10]. Since their proposal, Transformers have been growing steadily due to their wide-ranging modelling abilities in several application fields such as computer vision, machine translation, reinforcement learning, time-series analysis for classification and prediction, etc. [11]. In particular, in the present study, we propose a Transformer network based on a two-branch (temporal and channel modules) architecture with long short-term memory (LSTM) recurrent layers, Gaussian RANGE ENCODING (GRE), a multi-head self-attention mechanism, and a block-recurrent transformer layer (Fig. 3). TypeFormer is able to map slices of keystroke sequences into a feature embedding space where representations of sequences belonging to the same subject (intra-subject variability) are closer than those belonging to different subjects (inter-subject variability). TypeFormer is trained with the triplet loss function, and the similarity of the feature embeddings is measured with Euclidean distance.

In this way, while subjects type freely on their devices, TypeFormer might verify their identities passively by comparing and processing continuously acquired data samples with previously acquired and processed enrolment data (Fig. 1).

In brief, the main contributions of the current work are as follows:

- We propose TypeFormer, a novel Transformer architecture for biometrics keystroke free-text verification (Fig. 3).
- We provide an analysis of the different modules that compose the final architecture, starting from the original Vanilla Transfomer, first considering only the temporal module (with and without the recurrent layers), then the channel module only, to reach the final configuration of TypeFormer;
- We perform an in-depth comparison with recent state-of-the-art keystroke verification systems based on LSTM recurrent neural networks (RNN) and Transformers. By replicating the experimental protocol and adopting the same dataset [12], we outperform previous approaches [13, 14] in terms of equal error rate (EER), i.e. 3.25% using only five enrolment sessions consisting in 50-keystroke sequences. As a result, we also reduce the traditional performance gap existing between mobile free-text and desktop fixed-text scenarios. Finally, we also analyse the behaviour of the model with different experimental configurations such as the length of the keystroke sequences and the amount of enrolment sessions.
- We make our experimental framework available to the research community, aiming to contribute to advancing the state of the art of keystroke biometrics[2].

The remainder of the article is organised as follows: Sect. 2 describes key aspects of keystroke and Transformers. Then, Sect. 3 presents the architecture of TypeFormer. The main characteristics of the databases considered are reported in Sect. 4. In Sect. 5, a detailed description of the experimental setup is reported. Section 6 contains the experimental results and the comparison with the state of the art. Finally, in Sect. 7, we sum up our contributions and expose future research lines.

2 Related works

2.1 Keystroke biometrics

Raw keystroke data generally consist in the timestamps of the actions of pressing and releasing a key, the key code typed, and additional features depending on the specific acquisition device such as the pressure and the area size of the finger. From the raw data, several features are commonly extracted:

[1] In contrast with *physiological* biometrics, which pertains to the biological characteristics of an individual, such as face or fingerprint, all means that enable or contribute to differentiating between individuals throughout the way they perform activities are labelled as behavioural, i.e. gait, keystroke dynamics, handwritten signature, etc.

[2] https://github.com/BiDAlab/TypeFormer.

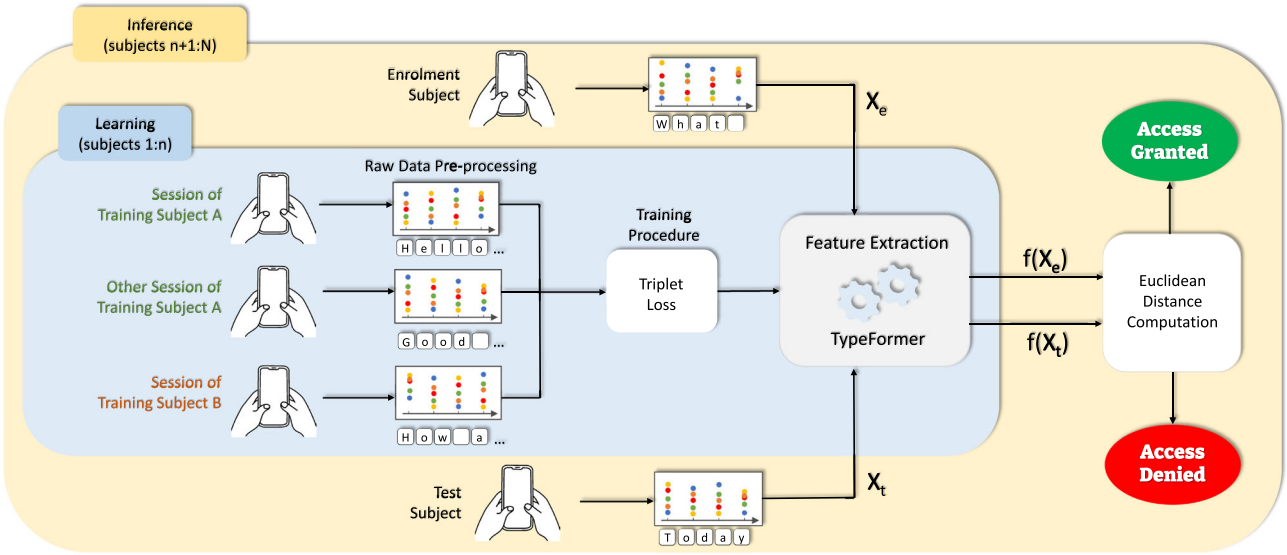

Fig. 1 Graphical representation of the workflow of TypeFormer, the proposed biometric keystroke free-text verification system

- Latencies, i.e. the time intervals of press-to-press, press-to-release (which is also known as the *hold time*), release-to-release, and release-to-press (*fly time*) events.
- Frequencies, such as the number of times per second a key is pressed or released.
- Error rates, related to the usage of backspaces or deletion options.
- Screen coordinates (x, y) and their displacement, angles, velocity, acceleration, etc.

Moreover, a typical classification of the keystroke systems is based on the text format [15]: fixed text (also known as text-dependent), in which the sequences of the keys typed by the user are pre-determined, as in the case of login credentials, and free text (text-independent), in which the sequences of keys typed are arbitrary, as in the case of messages. The latter entails additional challenges in comparison with the former, i.e. the unstructured and sparse nature of the information captured, more frequent typing errors, and differences in between enrolment and verification sessions, leading to a higher intra-subject variability. The performance might also be affected if the same subject is able to speak different languages [16]. As a result, the performance reachable in the free-text scenario is usually worse than in the case of the fixed-text one [13].

Although biometric recognition based on keystroke has been investigated for over a decade [17, 18], it can be still considered a biometric modality at the early stages, especially for mobile devices. In fact, before their application to mobile touchscreens, keystroke dynamics has been studied on the mechanical keyboards of desktop and laptop computers, for which, up to date, more in-depth evaluations have been conducted, and commercial applications have been proposed [17]. In addition, mobile devices entail further challenges with respect to desktop ones, such as the unconstrained and non-stationary acquisition conditions, possibly due to the users' activity, body position, emotional state, etc. [19].

We describe next some of the key factors in the development and evaluation of a keystroke dynamics system:

- Authentication performance, quantified through popular metrics in the field of biometrics, such as EER, false acceptance rate (FAR), false rejection rate (FRR), true acceptance rate (TAR), accuracy, area under the curve (AUC), etc.
- Number of data subjects included in the database for development and evaluation of the technology.
- Amount of data required for each subject, i.e. number and duration of enrolment and verification sessions.
- Text format: fixed text, transcript, or fully free text.
- Time interval between two acquisition sessions of the same subject, which can be a major source of variability due to biometric ageing, as observed in other behavioural biometric modalities [20].
- Information acquired, such as the timestamps of the actions of pressing and releasing a key, the key code typed, and additional features depending on the specific acquisition device such as the pressure.
- Instructions given to the subject during data acquisition which can lead to a restricted acquisition environment.
- Other parameters such as the memory required to store and deploy the model, prediction time, etc.

A typical issue of the field of keystroke biometrics is the heterogeneity of databases, experimental protocols, and metrics. Therefore, a rigorous comparison between the different performance values is a difficult operation. To

alleviate this aspect, Morales et al. provided a common experimental framework for the fixed-text format by presenting the Keystroke Biometrics Ongoing Competition (KBOC) for user authentication using keystroke biometrics [21].

2.2 Biometric keystroke verification

This section provides an overview of the key aspects of previous keystroke verification systems presented in the literature. The discussed studies are also reported in Table 1 in chronological order. We consider systems developed in both desktop ($\mathcal{D}$) and mobile ($\mathcal{M}$) scenarios.

2.2.1 Traditional approaches

In one of the earliest pioneering works on keystroke biometrics [22], Monrose and Rubin proposed a free-text keystroke algorithm by using the mean latency and standard deviation of digraphs and computing the Euclidean distance between each test sequence and the reference profile. Gunetti and Picardi [23] then extended the previous algorithm to n-graphs. More recently, due to their popularity, similar methods were used in [35] (2015) to study the effect of the data size on the performance of free-text keystroke, in [41] (2017) to study how detecting the user's position before authentication can significantly improve performance, and in [43] (2017) for benchmarking the large-scale database published, the Clarkson II database. The inclusion of time-related features such as rhythm and tempo was proposed in [28]. The random forest (RF) classifier was adopted in [53] to assess which are the most significant features of digraph-based algorithms (2020).

A very popular method for keystroke biometrics is support vector machine (SVM). Following the previous findings, in [30] and [38], combinations of the existing digraphs method for feature extraction and a SVM classifier to authenticate users were proposed. SVM was also adopted in [29] and in [33] in conjunction with mobile device background sensor data. Regardless of the classifier used, fusing keystroke dynamics with simultaneous movement sensor data included in mobile devices has proved to be very beneficial in terms of authentication results [5, 9, 52]. In a broad study (2018), Cilia et al. [49] studied how differentiating typing modes (one or two hands) and user activity (standing or moving) during the development of a keystroke verification system based on SVM can improve the authentication performance significantly.

Among other classifiers, we mention Hidden Markov Models (HMM), used in [24] to exploit typing rhythms in keystroke dynamics, and then extended by Monaco et al. [44] into Partially Observable Hidden Markov Models (POHMM). With k-Nearest Neighbour (k-NN) [25] and fuzzy logic [27], promising results have also been achieved in the early days of mobile keystroke biometrics. In the same epoch (2009), Killourhy and Maxion collected one of the first public databases of the field, the CMU keystroke dynamics database, and they carried out a benchmark evaluation with 14 different algorithms including Manhattan, Euclidean, and Mahalanobis distances, k-Nearest Neighbour, SVM (one-class), a neural network, fuzzy logic, and k-means [26]. A similar benchmark study was conducted in [42] on several algorithms such as Gaussian and Parzen Window Density Estimation, one-class SVM, k-NN, and k-means.

2.2.2 Deep learning approaches

The advent of DL-based systems has not spared the field of keystroke biometrics, improving significantly the authentication performance, in particular in the more challenging free-text scenario. In [31] (2013), it was shown that a deep neural network was capable of outperforming other algorithms on the CMU keystroke dynamics database [26]. Approaches based on neural networks were also used for complementary tasks to improve the authentication performance, such as predicting the digraphs that are not present among the enrolment sessions by analysing the relation between the keystrokes [32]. In [39], a convolutional neural network (CNN) was introduced in combination with a Gaussian data augmentation technique for the fixed-text scenario, while in [34], a neural network was applied to RGB histograms obtained from fixed-text keystroke data. Moreover, multi-layer perceptron (MLP) architectures have also been explored [58] ($\mathcal{M}$).

In [50], based on the observation that a RNN is a very suitable structure to learn from time-series [60, 61], a combination of a convolutional and a recurrent network was proposed in order to extract higher level keystroke features on the SUNY Buffalo database [51] (2019). The convolution process is performed before feeding the sequence to the recurrent network to characterise the keystroke sequence better. RNN variants are popular in keystroke biometrics, such as in [55] (birectional RNN) or in [59] ($\mathcal{M}$), in which keystroke sequences are arranged as an image-like matrix and then processed by a CNN combined with a gated recurrent unit (GRU) network. In 2021, Acien et al. presented TypeNet [13], a Siamese LSTM RNN for free-text keystroke biometrics. They considered the largest public databases to date, collected by researchers from the Aalto University, [54], and [12], with, respectively, around 168,000 and 68,000 subjects of free-text keystroke data divided into 15 acquisition sessions per subject. In their wide-ranging work, among other things, they achieved state-of-the-art authentication results at large scale in terms

Table 1 Summary of different approaches presented in the literature for keystroke dynamics verification

Study	Database (Public)	Number of subjects	Scenario	Classifier[1]	Performance [%]	Text format	Data amount
Monrose and Rubin [22]	Self-collected (✗)	42	$\mathcal{D}$	Weighted Euclidean dist	90.7 (Acc.) for Fixed Text 23.0 (Acc.) for Free Text	Fixed, free	Few sentences
Gunetti and Picardi [23]	Self-collected (✗)	205	$\mathcal{D}$	Different distance measures	< 0.005 (FAR), < 5 (FRR)	Free	700–900 characters
Jiang et al. [24]	Self-collected (✗)	58	$\mathcal{D}$	HMM	2.54 (ERR)	Fixed	20 strokes on average
Saevanee et al. [25]	Self-collected (✗)	10	$\mathcal{M}$	k-NN	99.0 (Accuracy)	Fixed	10-digit numbers
Killourhy and Maxion [26]	CMU database (✔)	51	$\mathcal{D}$	Manhattan dist., k-NN, SVM, Mahalanobis, NN, Euclidean dist., FL, *k*-means	0.096 (EER) with Manhattan dist.	Fixed	10 keystrokes
Zahid et al. [27]	Self-collected (✗)	25	$\mathcal{M}$	FL, PSO	2.07 (FAR), 1.73 (FRR)	Fixed	250 keystrokes
Hwang et al. [28]	Self-collected (✗)	25	$\mathcal{M}$	FF-MLP, RBFN, NN	4 (EER)	Fixed	4 digits
Giot et al. [29]	GREYC Web-based (✔) [29]	100	$\mathcal{D}$	SVM	15.28 (EER)	Fixed	5 captures
Balagani et al. [30]	Self-collected (✗)	34	$\mathcal{D}$	SVM	< 1 (Average Error Rate)	Free text	500 keystrokes
Deng and Zhong [31]	CMU database (✔) [26]	51	$\mathcal{D}$	GMM, NN	3.5−5.5 (EER)	Fixed, free	1 sequence
Ahmed et al. [32]	Self-collected (✔)	53	$\mathcal{D}$	Neural network	Controlled: 2.13 (EER, 0 FAR, 5 FRR) Uncontrolled: 2.46 (EER, 0.01 FAR, 4.8 FRR)	Free	500 actions
Gascon et al. [33]	Self-collected (✗)	300	$\mathcal{M}$	SVM	92 (TAR at 1% FAR)	Free	160 keystrokes
Alpar [34]	Self-collected (✗)	10	$\mathcal{D}$	NN, RGB histograms	90 (Acc.)	Fixed	15 characters
Huang et al. [35]	Clarkson I (✔) [36]	39	$\mathcal{D}$	Same as [23]	$\sim$ 1 (Impostor Pass Rate)	Free	1 k–10 k keystrokes
Morales et al. [21]	BiosecurID (✔) [37]	300	$\mathcal{D}$	Manhattan	5.32 (EER)	Fixed	$\sim$25 keystrokes
Çeker and Upadhyaya [38]	Clarkson I (✔) [36]	34	$\mathcal{D}$	SVM	$\sim$0 (EER)	Free	500 keystrokes
Çeker and Upadhyaya [39]	CMU database (✔) [26], GREYC Keystroke (✔) [40], GREYC Web-Based (✔) [29]	267	$\mathcal{D}$	CNN	2.02 (EER)	Free	Few keystrokes
Crawford et al. [41]	Self-collected (✗)	36	$\mathcal{M}$	Decision Tree	> 93 (AUC)	Free	Few keystrokes
Kim et al. [42]	Self-collected (✗)	150	$\mathcal{D}$	GDE, PWDE, 1-SVM, *k*-NN, and *k*-means	(EER: 0.44 for Korean, 0.84 for English)	Free	100–1000 keystrokes
Murphy et al. [43]	Clarkson II (✔) [43]	103	$\mathcal{D}$	Same as [23]	2.17−10.7 (EER)	Free	1000 keystrokes
Monaco et al. [44]	CMU database (✔) [26], (✔) [45], (✔) [46], (✔) [47], (✔) [48]	$\sim$50	$\mathcal{D}$	POHMM	0.6–9 (EER), 60.7−97.1 (Accuracy)	Fixed, free	0.12−55.18 events (on average)
Cilia et al. [49]	Self-collected (✔)	24	$\mathcal{M}$	SVM	0.44−3.93 (EER)	Fixed	Sentence based

Table 1 (continued)

Study	Database (Public)	Number of subjects	Scenario	Classifier[1]	Performance [%]	Text format	Data amount
Lu et al. [50]	SUNY buffalo (✔) [51], Clarkson II (✔) [43]	75	$\mathcal{D}$	CNN + RNN	2.67 (EER)	Free	30 keystrokes
Kim et al. [52]	Self-collected (✔)	50	$\mathcal{M}$	KS stat	< 0.05 (EER)	Free	$\sim$200 keystrokes
Ayotte et al. [53]	SUNY Buffalo (✔) [51], Clarkson II (✔) [43]	101, 148	$\mathcal{D}$	RF	7.8 (EER)	Free	200 digraphs
Acien et al. [13]	Aalto databases (✔) [12, 54], SUNY Buffalo (✔) [51], Clarkson II (✔) [43]	168 K	$\mathcal{D}, \mathcal{M}$	RNN	9.2 (EER) for $\mathcal{M}$, 2.2 for $\mathcal{D}$	Free	30–150 keystrokes
El-Kenawy et al. [55]	RHU dataset [56], MEU-Mobile KSD Dataset [57]	101, 148	$\mathcal{M}$	Bi-RNN	99.02 (Acc.), 99.32 (Acc.)	Fixed	Few keystrokes
Stylios et al. [58]	Self-collected (✔)	39	$\mathcal{M}$	MLP	97.18 (Acc.)	Fixed	$\sim$2 min sessions
Li et al. [59]	SUNY buffalo (✔) [51], Clarkson II (✔) [43]	101, 148	$\mathcal{D}$	CNN + RNN	97.68 (Acc.), 88.62 (Acc.)	Free	50 keystrokes
Stragapede et al. [14]	Aalto Database $\mathcal{M}$ (✔) [12]	60 K	$\mathcal{M}$	Transformer	3.84 (EER)	Free	50 keystrokes
TypeFormer	Aalto databases (✔) [12, 54], SUNY Buffalo (✔) [51], Clarkson II (✔) [43]	60 K	$\mathcal{D}, \mathcal{M}$	Transformer	3.25 (EER)	Free	30–100 keystrokes

[1] *HMM* = Hidden Markov Models, *k-NN* = k-Nearest Neighbours, *SVM* = Support Vector Machine, *NN* = Neural Network, *FL* = Fuzzy Logic, *PSO* = Particle Swarm, Optimisation, *FF-MLP* = Feed-Forward Multi-Layer Perceptron, *RBFN* = Radial Basis Function Network, *GMM* = Gaussian Mixture Model, *CNN* = Convolutional NN, *GDE* = Gaussian, Density Estimator, *PWDE* = Parzen Window Density Estimator, *POHMM* = Partially Observable HMM, *RNN* = Recurrent Neural Network, *KS* = Kolmogorov–Smirnov, *RF* = Random, Forest, *Bi-RNN* = Bidirectional RNN, and *MLP* = Multi-Layer Perceptron

of EER (%) while attempting to minimise the amount of data per subject required for enrolment. Following [13], in [14], in 2022, we presented a preliminary attempt to use a Transformer architecture for keystroke biometrics, outperforming TypeNet in a specific experimental setup. We selected [13] as a reference study for several reasons: (i) They adopt the largest mobile free-text keystroke databases available, the Aalto mobile keystroke database [12], (ii) their experimental protocol is publicly available on GitHub, allowing us to use the same sets of subjects and metrics, for development and evaluation, and (iii) they achieved state-of-the-art results for free-text mobile keystroke biometrics. Consequently, references [13] and [14] are particularly relevant to the current study as they use the same development and evaluation databases, and experimental protocol, allowing a direct comparison of the proposed systems (Sect. 6). Recently, in [62], a novel approach called DoubleStrokeNet for recognising subjects using bigram embeddings was proposed. DoubleStrokeNet considers a Transformer-based neural network that distinguishes between different bigrams. Additionally, self-supervised learning techniques were employed to compute embeddings for both bigrams and users. The authors experimented with the Aalto databases, reaching very competitive results in terms of recognition performance. Leveraging the temporal features of specific bigrams is a route of potential interest in modelling subjects' typing behaviour. It is difficult to compare results across different studies, which adopt different experimental settings, e.g. training and evaluation data.

2.3 Introduction to transformers

The first Transformer was proposed by Vaswani et al. as a new encoder–decoder architecture [10]. Such model, later nicknamed the Vanilla Transformer, is based purely on attention mechanisms, abandoning the idea of using convolutions or recurrence. The Vanilla Transformer was

proposed for the task of machine translation, achieving remarkable results in comparison with existing systems in terms of quality of text translation and time consumption. In comparison with existing DL architectures such as CNNs or RNNs, the main advantages of the Transformer can be summarised as follows: (i) All sequences are processed in parallel; (ii) a self-attention mechanism is introduced to deal with long sequences; (iii) the training is more efficient, modelling the whole sequences at once; and (iv) inspection of the whole sequences at once, without the need to summarise previous samples [10, 63, 64].

Later, several variations of the original Transformer architecture have been proposed to overcome some of its drawbacks and to deploy it in other application fields. In fact, its quadratic computational complexity and its considerable memory usage limited its application to longer time-series signals. To alleviate these aspects, the Two-stream Convolution Augmented Human Activity Transformer (THAT) was proposed by Li et al. for the task of human activity recognition (HAR) [65]. Such architecture was designed based on the assumption that, similarly to images, time-series signals have information in two dimensions. Therefore, the model comprises two modules: (i) the temporal module (extracting time features from unchanged data) and (ii) the channel module (extracting channel features from transposed data). Then, the features extracted by each of the modules are concatenated for the prediction task. Another example of an interesting Transformer architecture variation is given by the block-recurrent transformer, that has been recently introduced by Hutchins et al. for the task of auto-regressive language modelling [64]. In this approach, thanks to the recurrent on series-wise connexions, all previous temporal information is retained. Furthermore, two attention mechanisms are applied at the same time (full- and cross-attention).

In the light of these and other adaptations, the popularity of Transformers increased in the past years due to the remarkable results obtained in other fields such as computer vision, reinforcement learning, time-series analysis for classification and prediction, biometrics, etc. [11, 66]. Recently, this lead to the emergence of large pre-trained Transformers, also referred to as foundation models (FMs), renowned for their adaptability across diverse tasks. These expansive pre-trained Transformers encompass varied architectures tailored to specific tasks, including large language models (LLMs) for natural language processing [67], vision Transformers (ViT) for visual tasks [68], and multimodal Transformers for tasks involving multiple modalities. Despite the widespread adoption of these sophisticated models, within the realm of behavioural biometrics, a scarcity of data presents a significant challenge, thereby limiting the evaluation of these models in this specific task domain within existing literature. A thorough discussion of Transformers in different domains is out of the scope of the current article. Nevertheless, we recommend two excellent surveys about vision Transformers [68] and Transformers for time-series [69].

A preliminary version of this work was published in [14] as the first application of Transformers to keystroke biometrics. This article significantly improves [14] in the following aspects: (i) We propose a new Transformer architecture, TypeFormer, leading to an improvement of the authentication performance; (ii) we provide a more extensive evaluation of the model, analysing the behaviour of the system with different experimental conditions such as the number of enrolment sessions and the length of the keystroke sequences; and (iii) we provide an in-depth analysis of state-of-the-art keystroke verification systems, remarking key aspects such as the scenario (fixed or free text) and database considered, classifier, and performance.

3 Proposed system: TypeFormer

This section contains a detailed description of all aspects of the proposed keystroke verification system.

3.1 Feature extraction

The raw keystroke information available consists essentially in the timestamp of the event of pressing (finger down) and releasing (finger up) a key, together with the ASCII code typed. Such data are processed to extract a set of five features per character typed:

[hold latency, inter-key latency, press latency, release latency, key pressed]

The above-mentioned features are shown in Fig. 2. Due to the fact that the length of the free-text sequences is not fixed, they are sliced or zero-padded to produce a fixed-size input, $(L = 30, 50, 70, 100)$, depending on the specific experiment (see Sect. 5). The ASCII code (key pressed) is normalised in the range [0, 1].

3.2 TypeFormer architecture

Following the same idea presented in [65], TypeFormer contains two modules, each of them in a specific branch, to which the pre-processed Transfomer input sequences X (Sect. 3.1) are fed (Fig. 3): a temporal module (temporal-over-channel features) and a channel module (channel-over-temporal features). In both channels, X is modelled using a GRE to preserve the information position. The output sequence is defined by an L_1 normalised vector representing the probability density function (PDF) of the Gaussian distributions G. Moreover, the final GRE is calculated by a weighted multiplication over several ranges,

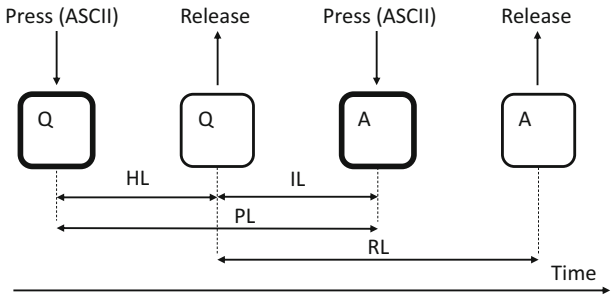

Fig. 2 Example of the keystroke features extracted from the Aalto mobile keystroke database [12]. *HL* hold latency; *IL* inter-key latency; *PL* press latency; *RL* release latency; and ASCII: key pressed

containing the behaviour of each of the samples in a different scenario.

The temporal module contains three ordered sets of layers. Each of the sets of layers is composed, respectively, by N, R, and M layers. The N and M layers are identical and made of two sub-layers: a multi-head self-attention mechanism and a multi-scale keystroke LSTM RNN layer. The multi-head self-attention mechanism connects the samples among the whole sequence obtaining long-range dependencies. The mechanism applies a weighted sum of the different values V over the different queries Q and the matching keys K. The output of the self-attention sub-layer is the result of applying the attention mechanism to F independent heads. Then, the multi-scale keystroke LSTM RNN layer is activated by ReLU functions. Each of the scales contains a unique kernel. Following each sub-layer, a residual connexion and a layer normalisation are included (*Add & Norm* in Fig. 3).

Between the N and M layers, R recurrent layers are included (graphically represented in detail on the right side of Fig. 3). The structure of such layers is based on the block-recurrent transformer architecture presented in [64]. Initially, the input sequence is shaped by a positional encoding. Then, a recurrent form of attention is introduced in the vertical and horizontal directions, based on two sub-layers in each of the directions: (i) a multi-head self-attention mechanism, which applies full-attention to the sequences to obtain the matching values V and keys K, and cross-attention to the current states (initialised to 0) to extract the queries Q (replicated in F independent heads); and (ii) a multi-scale keystroke CNN network, which comprises a CNN with ReLU activations and unique kernels for each of the scales. Every sub-layer is preceded by a layer normalisation and followed by a residual connexion (Add & Norm). While the multi-scale keystroke CNN network remains unchanged, the multi-head self-attention mechanism applies cross-attention to the sequences to obtain the matching queries Q, and full-attention to the current states to extract the keys K and the values V (such mechanism is replicated in F independent heads). Furthermore, the residual connexions are replaced by forget gates, altering the current states.

The channel module input sequence X is transposed and modelled by the GRE. Then, H layers (analogous to the N and M layers of the Temporal Module) are included, followed by a residual connexion and a layer normalisation (Add & Norm).

Subsequently, each of the modules is followed by a convolutional layer, after which the similarity of the output features is concatenated into an output vector P and fed into a sigmoid layer. Finally, for the authentication task considered in the present study, the output feature embedding vectors are compared using the Euclidean distance.

The architecture of TypeFormer is based on a preliminary transformer version proposed in [14]. However, this architecture has been modified leading to improved

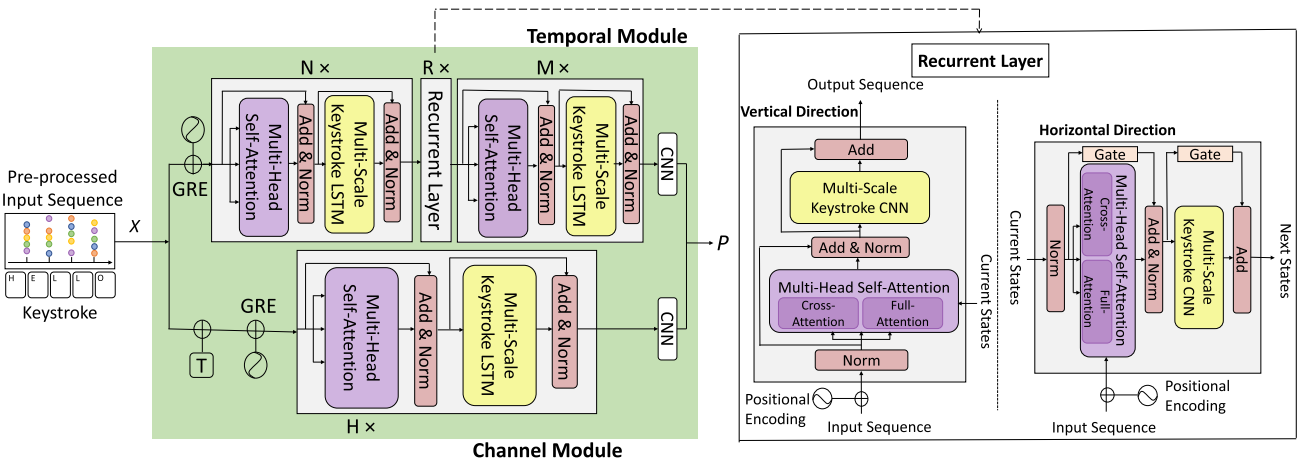

Fig. 3 Graphical representation of TypeFormer, based on a Transformer architecture for biometrics keystroke free-text verification. *T* transposition operation; *GRE* Gaussian range rncoding; N, R, M, H: Number of layers of each of the modules; X Pre-processed input sequence; and P Output feature embedding vector

biometric recognition performance. In particular, the main changes can be summarised as follows: *(i)* The convolutional layers in the temporal and channel modules have been changed to LSTM RNN layers, which show better ability to model time-domain signals [61]; and *(ii)* in the temporal module, a set of block-recurrent transformer layers based on [64] has been included between two sets of identical recurrent layers, following the idea of [66]. The block-recurrent block introduces a recurrent form of attention, in alternative to using the dot-product or periodicity-based series mechanism, which fix an attention window size, summarising the sequence that the model has previously seen. As presented in Table 2, this leads to improved recognition results.

The specific details of the hyperparameter implementation for the proposed Transformer are described in Sect. 5.1.

4 Databases description

4.1 The database

The Aalto mobile keystroke database is a large-scale database for mobile keystroke biometrics involving around 260,000 subjects [12]. In this work, we have selected all subjects that completed at least 15 acquisition sessions, reducing the number of subjects to 62,454. The raw data available in the Aalto mobile keystroke database consist in the timestamps of the key press (finger down) and key release (finger up) gestures with a 1-ms resolution. The data were captured through a mobile web application in an unsupervised way. Subjects were asked to read, memorise, and type in their smartphone English sentences that were randomly selected from a set of 1525 sentences obtained from the Enron mobile mail [70] and the Gigaword Newswire corpora [71]. Therefore, the text format adopted is free text, with sentences containing at least three words or 70 characters. Moreover, the volunteers were asked to type as fast and accurately as possible. Concerning the volunteers, they were selected from 163 countries, approximately 68% of the subjects involved were English native speakers, and around 31% of them took a typing course.

5 Experimental protocol

5.1 TypeFormer hyperparameters

The best configuration found in terms of the hyperparameters of the proposed Transformer is described below. To achieve this, several combinations of hyperparameter were adopted for different trainings. Then, the EER on the validation set was used to select the best model among all trainings. The Gaussian range encodings contain $G = 20$ Gaussian distributions. The temporal module comprises $N = 9$, $R = 2$, and $M = 1$ layers with $F = 10$ heads each, while the channel module $H = 1$ layer with $F = 5$ heads. In both modules, the multi-scale keystroke LSTM contains three recurrent layers with kernel sizes 1, 3, and 5, respectively. Each of them comprises D units and ReLU activation functions, followed by dropout layers with a rate of 0.1. The multi-scale keystroke CNN networks of the R recurrent layers contain D units each (where D corresponds to the keystroke sequence length L), ReLU activation functions, and kernel sizes 1, 3, and 5, respectively, followed by dropout layers with a rate of 0.1. Subsequent to the temporal and channel modules, two convolutional layers are included with D units, ReLU activation functions, and kernel sizes 128 and 32, respectively. Each of the convolutional layers is followed by dropout layers with a rate of 0.5. Finally, a max-pooling layer followed by a linear layer with sigmoid activation function is included. The final output vector contains $S = 64$ features.

5.2 Model development

In order to perform a fair comparison across different DL architectures, in the current work, we replicate the public experimental protocol presented by Acien et al. in [13]. Specifically, data belonging to the same non-overlapping 30,000 and 400 subjects have been used, respectively, for the purpose of training and validation. Each subject data are organised into 15 acquisition sessions. The triplet loss function is employed for the training, and a margin of $\alpha = 1.0$ was set on top of the Euclidean distance for each of the pair combinations in the triplet. Additionally, the Adam optimiser with a learning rate of 0.001 is used. The Transformer is trained for 1000 epochs, considering

Table 2 Experimental results of the different modules implemented in the development of TypeFormer, in comparison with the Vanilla Transformer [10] (E is the number of enrolment sessions)

System	$E = 1$	$E = 2$	$E = 5$	$E = 7$	$E = 10$
Vanilla Transformer [10]	10.28	8.56	7.41	6.95	6.61
Temporal Module w/o Rec. layer	8.15	6.43	5.12	4.73	4.29
Temporal module w/ Rec. layer	7.12	5.49	3.94	3.63	3.15
Channel module	17.29	15.50	13.54	13.07	12.55
TypeFormer (Temp. + Channel Module w/ Rec. Layer)	6.17	4.57	3.25	2.86	2.54

Table 3 Intra-database evaluation: system performance results in terms of EER for the final evaluation dataset of the Aalto mobile database

Sequence length L	System	Number of enrolment sessions E				
		1	2	5	7	10
30	Acien et al. [13]	14.20	12.50	11.30	10.90	10.50
	TypeFormer	9.48	7.48	5.78	5.40	4.94
50	Acien et al. [13]	12.60	10.70	9.20	8.50	8.00
	Preliminary Transformer [14]	6.99	–	3.84	–	3.15
	TypeFormer	6.17	4.57	3.25	2.86	2.54
70	Acien et al. [13]	11.30	9.50	7.80	7.20	6.80
	TypeFormer	6.44	5.08	3.72	3.30	2.96
100	Acien et al. [13]	10.70	8.90	7.30	6.60	6.30
	TypeFormer	8.00	6.29	4.79	4.40	3.90

roughly 30,000 triplets per epoch, arranged into 1024-sequence-sized batches. The triplets are formed by sampling subjects randomly and with uniform distribution across the training set. At the end of each training epoch, the model performance is quantified in terms of EER, and according to such metric, the best model is selected to be tested on the final evaluation subset. TypeFormer is implemented in PyTorch.

5.3 Model evaluation

We describe next the experiments considered in the present study to validate the proposed TypeFormer. In all of them, different subjects are used for training and evaluating the keystroke verification model.

The first experiment analyses the performance of TypeFormer over an evaluation set of $U = 1,000$ unseen subjects obtained from the same database considered in training. At the end of each of the training epochs, the best model is selected using a separate validation subset. We follow the same protocol as [13], considering E enrolment sessions per subject. The genuine and impostor score distributions are subject-specific. For each subject, genuine scores are obtained comparing the enrolment sessions (E) with five verification sessions. The Euclidean distances are computed for each of the verification sessions with each of the E enrolment sessions, and then, values are averaged over the enrolment sessions. Therefore, for each subject, there are five genuine scores, one for each verification session. Concerning the impostor score distribution, for every other subject in the evaluation set, the averaged Euclidean distance value is obtained considering one verification session and the above-mentioned five enrolment sessions. Consequently, for each subject, there are 999 impostor scores. Based on such distributions, the EER score is calculated per subject, and all EER values are averaged across the entire evaluation set. The number of enrolment sessions is variable ($E = 1, 2, 5, 7, 10$) in order to assess the performance adaptation of the system to reduced availability of enrolment data. Additionally, also the experiments are repeated changing the input sequence length, $L = 30, 50, 70, 100$, to evaluate the optimal keystroke sequence length.

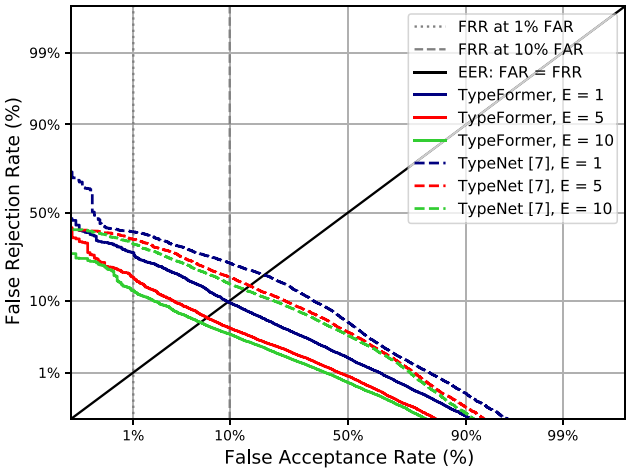

Fig. 4 DET curves comparing the performance of TypeFormer with TypeNet ([13]) for keystroke sequences of length $L = 50$. E corresponds to the number of enrolment sessions considered. The solid black line $y = x$ corresponds to all possible EER points (for which FAR = FRR by definition), whereas the grey lines, respectively, represent the FRRs at 1% FAR (dotted) and FRRs at 10% FAR (dashed)

6 Experimental results

Starting from the initial vanilla transformer proposed in [10], to validate each part of final proposed system, Table 2 presents the experimental results of the different modules implemented in the development of TypeFormer. The results are obtained on the final evaluation dataset of the Aalto mobile database. This analysis is carried out by considering a variable number of enrolment sessions $E = 1, 2, 5, 7, 10$ along the columns and sequence length

Table 4 Global EER (%), FRR at 1% FAR (%), and FRR at 1% FAR (%) of TypeNet [13] and TypeFormer for different amounts of enrolment sessions E. Such values correspond to the intersection points of the DET curves with the straight lines plotted in Fig. 4. The sequence length $L = 50$

Enrolment **sessions**	System	Global EER (%)	FRR at 1% FAR (%)	FRR at 10% FAR (%)
$E = 1$	Acien et al. [13]	18.20	38.99	23.19
	TypeFormer	9.72	27.97	9.53
$E = 5$	Acien et al. [13]	14.40	34.93	17.40
	TypeFormer	6.32	17.47	4.68
$E = 10$	Acien et al. [13]	13.16	32.42	14.91
	TypeFormer	5.52	12.81	3.85

Table 5 Comparison of the performance achieved by the proposed TypeFormer with related systems that followed different experimental protocols in the studies in which they were originally proposed (E = number of enrolment sessions = 5 and L = number of enrolment sessions considered = 50)

System	EER (%)
POHMM [72]	40.40
Digraphs [38]	29.20
CNN+RNN [50]	12.20
TypeNet [13]	9.20
Preliminary Transformer [14]	3.84
TypeFormer	3.25

$L = 50$. Although the Vanilla Transformer is solely based on attention mechanisms, it shows the effectiveness of the Transformer architecture in modelling keystroke sequences. First, this architecture is modified by including the Gaussian range encoding (instead of the Positional Encoding originally used in the Vanilla Transformer). Then, the point-wise feed-forward networks of the Vanilla Transformer are changed with LSTM recurrent layers (Temporal w/o Rec. Layer). By doing so, we obtain an improvement for all considered amounts of enrolment sessions, and the recognition performance in terms of EER is improved on average by a 28.70%. Following [64], a block-recurrent transformer layer is introduced in the temporal module in the case of the temporal with recurrent layer configuration. This further reduces the EER by a 20.03% (Temporal w/ Rec. Layer). Finally, we considered the combination of the temporal with recurrent layer and channel module configurations, corresponding to the final TypeFormer architecture.

Table 3 shows the results achieved by TypeFormer considering different sequence lengths L. In addition, to provide a better comparison of TypeFormer with recent state-of-the-art keystroke biometric systems, we include the results achieved by TypeNet in [13] and our preliminary study [14] on the same dataset as shown in the previous Table 2. In general, in Table 3, we can see that in all cases, TypeFormer outperforms previous approaches over the same evaluation set of 1000 subjects. In particular, the performance improvement of TypeFormer averaged over all cases in the table ($E = 1, 2, 5, 7, 10$ and $L = 30, 50, 70, 100$) consists in 47.3% in relative terms with respect to TypeNet [13], an LSTM RNN-based system.

Additionally, considering only the results of Table 3 obtained by TypeFormer, it is possible to observe that in all cases, the EER values decrease as the number of enrolment sessions E increases. Such trend is predictable and consistent for all sequence lengths L. Also, the rate of improvement is higher going from $E = 1$ to $E = 5$ sessions (relative improvement of almost 50% going from 6.17% to 3.25% EER for $L = 50$) than from $E = 5$ to $E = 10$ (relative improvement of around 20% going from 3.25% to 2.54% EER for $L = 50$).

Similarly, by carrying out an analogous analysis along the rows, it is noticeable that increasing the input sequence length L from 30 to 50, there is a significant improvement (42.64% in relative terms on average over all considered enrolment session amounts E) in terms of EER. Nevertheless, such trend is reversed when increasing the sequence length L to 70 or 100 (respectively, a performance degradation of 12.38% and 28.38% in relative terms on average over all considered enrolment session amounts E), leading to the conclusion that the optimal sequence length must be around 50. This could be due to the fact that the zero-padding operation carried out to equalise the length of different keystroke sequences is not beneficial for the Transformer-based architecture that relies on an attention mechanism, that can perhaps be optimised. In case of the RNN-based reference system [13], the longer the input sequences, the better the results, showing the beneficial effects of the masking layer included in their network.

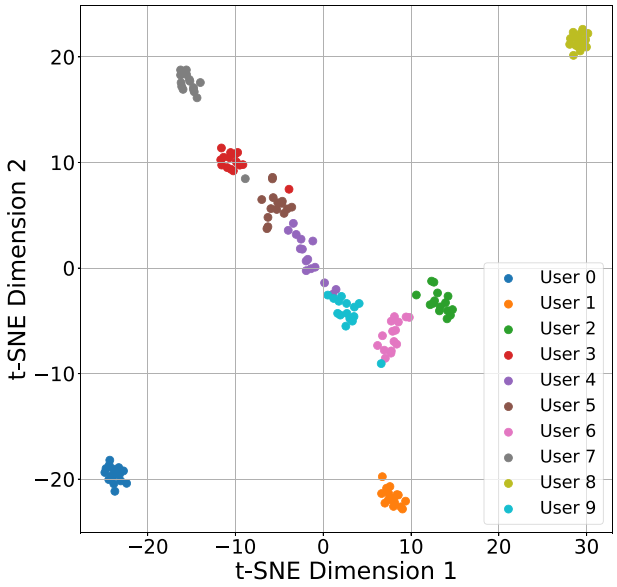

Fig. 5 Two-dimensional graphical visualisation of the latent space through t-SNE considering 15 sessions of 10 subjects [73]. Selected parameters: `perplexity` = 14, `init` = 'pca', `n_iter` = 1000

To provide a graphical representation of the differences in the performance of the compared systems, Fig. 4 reports the detection error trade-off (DET) curves computed for the different number of enrolment sessions available ($L = 50$). The graph shows that our proposed approach outperforms the LSTM RNN of TypeNet in all cases, i.e. $E = 1$ (TypeFormer) enrolment session vs. $E = 10$ (TypeNet). This shows the ability of TypeFormer to model keystroke dynamics. The DET curves shown in Fig. 4 are plotted considering the entire *global* genuine and impostor score distributions, i.e. by grouping all scores regardless of the specific subject. The solid black line $y = x$ corresponds to all possible EER points (for which $FAR = FRR$ by definition), whereas the grey lines, respectively, represent the FRRs at 1% FAR (dotted) and FRRs at 10% FAR (dashed). For $E = 5$, TypeFormer achieves 6.32% of global EER, while, by shifting the system threshold to set a FAR of 1% and 10%, we obtain corresponding FRRs of, respectively, 17.47% and 4.68%. Table 4 contains all the intersection points obtained from Fig. 4. We observe that setting the system threshold to a high security level (corresponding to $FAR = 1\%$) affects the usability of the system, i.e. it increases the amount of false rejections for the legitimate users. The computation of such metrics is limited to the global scenario due to the higher amount of genuine scores.

Lastly, Table 5 presents a comparison of the proposed TypeFormer with other systems presented in the literature that were not originally evaluated according to the protocol adopted in this work [12]: digraphs and SVM [38], POHMMs [72], and a combination of RNNs and CNNs [50]. The evaluation of the different system takes place on the same set of 1000 subjects considering $E = 5$ and $L = 50$. TypeFormer shows the best performance, with EER absolute improvements of 37.15% (POHMM [72]), 32.45% (Diagraphs [38]), 8.95% (CNN + RNN [50]), 5.95% (TypeNet [13]), and 0.59% (our preliminary Transformer architecture [14]). Such results show the potential of TypeFormer and Transformer-based architectures in the challenging free-text mobile scenario. For completeness, we also report the inference time for a single feature extraction instantiation. Specifically, we consider as input a biometric sample in the form of the pre-processed five features described in Sect. 3.1 and a keystroke sequence length $L = 50$. The inference time is 46.4 ms on average, considering all embeddings computed on the evaluation set.[3] The experiments are carried out on a NVIDIA GeForce RTX 3070 Ti graphics card. In terms of number of parameters, TypeFormer has approximately 1.8M, whereas the preliminary Transformer has 400K, and TypeNet has 200K.

6.1 Analysis of the feature embeddings

The output feature embeddings extracted by TypeFormer lie in a 64-dimensional space, and their pairwise relative positioning is measured throughout the Euclidean distance. In this scenario, mathematical methods like the popular t-SNE [73] are useful to visualise data points in such high-dimensional spaces. Figure 5 depicts the output feature embedding space reduced to two dimensions through t-SNE. For better visualisation, we include examples of 10 random subjects of the database (15 acquisition sessions per subject). Apart from few outliers, most groups are clearly separated, while data points belonging to the same subjects are closer together. This is an indicator of small intra-class variability and high inter-class variability.

7 Conclusions and future work

In the current article, we have proposed a novel Transformer-based architecture, TypeFormer, for the task of free-text mobile keystroke authentication. TypeFormer features two branches (temporal and channel modules) with long short-term memory (LSTM) layers, Gaussian range encoding (GRE), a multi-head self-attention mechanism, and a block-recurrent transformer layer, and it was trained with triplet loss. Its output consists in feature embedding vectors representing points in the output hyperspace. The distance between embedding vectors is

[3] `sklearn.manifold.TSNE` -- scikit-learn 1.1.1 documentation.

measured through the Euclidean distance, and it is less for instances of data belonging to the same subject than for ones of different subjects. The development of the model is based on the Aalto mobile keystroke database [12], the largest public databases of mobile keystroke dynamics. First, we have performed an analysis to validate the different modules that are present in the final presented Transformer architecture. Then, in order to compare TypeFormer with the highest-performing systems recently proposed in the literature, we have replicated the experimental protocol of two recent studies [13, 14], by varying the number of enrolment sessions ($E = 1, 2, 5, 7, 10$), input keystroke sequence lengths ($L = 30, 50, 70, 100$), and considering the same database repartition. In all cases, TypeFormer outperformed previous approaches, reaching as little as 3.25% EER considering $E = 5$ and $L = 50$. This would be an absolute improvement of 5.95% EER with respect to previous LSTM RNN-based model (the corresponding relative improvement is around 65%) [13]. To advance the state of the art of free-text mobile keystroke biometrics, we make our proposed approach and experimental framework public[4].

Concerning future work, the next directions of research will go towards exploring the effectiveness of Transformers in modelling other biometric traits [74], including data captured by mobile device sensors [9, 75] and synthetic data [76]. To this end, we will consider the optimisation of the Transformer architecture to improve the performance with longer sequences. Additionally, more sophisticated training approaches will be investigated, in terms of the loss function, such as the implementation of hard triplet mining, in order to force the model to learn from harder comparisons [77], and output feature embedding distance metrics. Finally, it would also be interesting to shed light on explainability and privacy aspects of mobile keystroke authentication [78, 79], i.e. investigating the subject information contained in the feature embeddings, i.e. gender, age, etc., to assess whether keystroke data should be treated as privacy-sensitive biometric data. For this, the Aalto mobile keystroke database can be useful due to the the subject metadata available.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 860315. Moreover, it has been supported by INTER-ACTION (PID2021-126521OB-I00 MICINN/FEDER) and Cátedra ENIA UAM-VERIDAS en IA Responsable (NextGenerationEU PRTR TSI-100927-2023-2).

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

[4] https://github.com/BiDAlab/TypeFormer.

Data availability The database used is freely available for download https://userinterfaces.aalto.fi/typing37k/. The database is associated to the publication [12]. In our GitHub repository https://github.com/BiDAlab/TypeFormer, we provide all the necessary information to replicate the experimental protocol of the benchmark evaluation of TypeFormer.

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organisation or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Thariq Ahmed HF, Ahmad H (2020) Device free human gesture recognition using Wi-Fi CSI: a survey. Eng Appl Artif Intell 87:103281
2. Rathgeb C, Tolosana Vera-Rodriguez R, Busch C (2022) Handbook Of digital face manipulation and detection: from deepfakes to morphing attacks. Springer, Berlin
3. ISO 9241-11:2018(en): Ergonomics of human-system interaction (2018) Part 11: usability: definitions and concepts
4. Patel VM, Chellappa R, Chandra D, Barbello B (2016) Continuous user authentication on mobile devices: recent progress and remaining challenges. IEEE Signal Process Mag 33(4):49–61
5. Stragapede G, Vera-Rodriguez R, Tolosana R, Morales A, Acien A, Le Lan G (2022) Mobile behavioral biometrics for passive authentication. Pattern Recognit Lett 157:35–41
6. Delgado-Santos P, Tolosana R, Guest R, Vera-Rodriguez R, Deravi F, Morales A (2022) GaitPrivacyON: privacy-preserving mobile gait biometrics using unsupervised learning. Pattern Recogn Lett 161:30–37
7. Delgado-Santos P, Stragapede G, Tolosana R, Guest R, Deravi F, Vera-Rodriguez R (2022) A survey of privacy vulnerabilities of mobile device sensors. ACM Comput Surv 54:1–30
8. Porwik P, Doroz R (2021) Adaptation of the idea of concept drift to some behavioral biometrics: preliminary studies. Eng Appl Artif Intell 99:104135
9. Stragapede G, Vera-Rodriguez R, Tolosana R, Morales A (2023) BehavePassDB: public database for mobile behavioral biometrics and benchmark evaluation. Pattern Recogn 134:109089
10. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A.N, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Proc. Adv Neural Inform Process Syst
11. Tay Y, Dehghani M, Bahri D, Metzler D (2022) Efficient Transformers: a survey. ACM Comput Surv

12. Palin K, Feit A.M, Kim S, Kristensson P.O, Oulasvirta A (2019) 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
13. Acien A, Morales A, Monaco JV, Vera-Rodriguez R, Fierrez J (2021) TypeNet: deep learning keystroke biometrics. behavior, and identity science. IEEE Trans Biomet 4(1):57–70
14. Stragapede G, Delgado-Santos P, Tolosana R, Vera-Rodriguez R, Guest R, Morales A (2023) Mobile keystroke biometrics using Transformers. In: proc. int. conf. on automatic face and gesture recognition
15. Mondal S, Bours P (2017) A Study on Continuous Authentication Using a Combination of Keystroke and Mouse Biometrics. Neurocomputing, 230: 1-22
16. Abuhamad M, Abusnaina A, Nyang D, Mohaisen D (2021) Sensor-based continuous authentication of smartphones' users using behavioral biometrics: a contemporary survey. IEEE Internet Things J 8(1):65–84
17. Maiorana E, Kalita H, Campisi P (2021) Mobile keystroke dynamics for biometric recognition: an overview. IET Biom 10(1):1–23
18. Roy S, Pradhan J, Kumar A, Adhikary DRD, Roy U, Sinha D, Pal RK (2022) A systematic literature review on latest keystroke dynamics based models. IEEE Access 10:92192–92236
19. Teh PS, Zhang N, Teoh ABJ, Chen K (2016) A survey on touch dynamics authentication in mobile devices. Comput Secur 59:210–235
20. Tolosana R, Vera-Rodriguez R, Fierrez J, Ortega-Garcia J (2019) Reducing the template ageing effect in on-line signature biometrics. IET Biom 8(6):422–430
21. Morales , Fierrez J, Gomez-Barrero M, Ortega-Garcia J, Daza R, Monaco J.V, Montalvão J, Canuto J, George A (2016) KBOC: Keystroke biometrics ongoing competition. In: proc. int. conf. on biometrics theory, applications and systems
22. Monrose F, Rubin A (1997) Authentication via keystroke dynamics. In: proc. conf. on computer and communications security
23. Gunetti D, Picardi C (2005) Keystroke analysis of free text. ACM Trans Inform Syst Secur 8(3):312–347
24. Jiang C.-H, Shieh S, Liu J.-C (2007) Keystroke statistical learning model for web authentication. In: proc. of the symp. on information, computer and communications security
25. Saevanee H, Bhatarakosol P (2008) User authentication using combination of behavioral biometrics over the touchpad acting like touch screen of mobile device. In: proc. int. conf. on computer and electrical engineering
26. Killourhy K.S, Maxion R.A (2009) Comparing Anomaly-detection algorithms for keystroke dynamics. In: proc. int. conf. on dependable systems networks
27. Zahid S, Shahzad M, Khayam S.A, Farooq M (2009) Keystroke-based user identification on smart phones. In: proc. int. workshop on recent advances in intrusion detection
28. Hwang S-S, Cho S, Park S (2009) Keystroke dynamics-based authentication for mobile devices. Comput Secur 28(1–2):85–93
29. Giot R, El-Abed M, Hemery B, Rosenberger C (2011) Unconstrained keystroke dynamics authentication with shared secret. Comput secur 30(6–7):427–445
30. Balagani KS, Phoha VV, Ray A, Phoha S (2011) On the discriminability of keystroke feature vectors used in fixed text keystroke authentication. Pattern Recogn Lett 32(7):1070–1080
31. Deng Y, Zhong Y (2013) keystroke dynamics user authentication based on gaussian mixture model and deep belief nets. International scholarly research notices
32. Ahmed AA, Traore I (2013) Biometric recognition based on free-text keystroke dynamics. IEEE Trans Cybern 44(4):458–472
33. Gascon H, Uellenbeck S, Wolf C, Rieck K (2014) Continuous Authentication on mobile devices by analysis of yyping motion behavior. Sicherheit 2014–Sicherheit, Schutz und Zuverlässigkeit
34. Alpar O (2014) Keystroke recognition in user authentication using ANN based RGB histogram technique. Eng Appl Artif Intell 32:213–217
35. Huang J, Hou D, Schuckers S, Hou Z (2015) Effect of data size on performance of free-text keystroke authentication. In: proc. int. conf. on identity, security and behavior analysis
36. Vural E, Huang J, Hou D, Schuckers S (2014) Shared research dataset to support development of keystroke authentication. In: proc. int. joint conf. on biometrics
37. Fierrez J, Galbally J, Ortega-Garcia J, Freire MR, Alonso-Fernandez F, Ramos D, Toledano DT, Gonzalez-Rodriguez J, Siguenza JA, Garrido-Salas J et al (2010) BiosecurID: a multimodal biometric database. Pattern Anal Appl 13(2):235–246
38. Çeker H, Upadhyaya S (2016) User authentication with keystroke dynamics in long-text data. In: proc. int. conf. on biometrics theory, applications and systems
39. Çeker H, Upadhyaya S (2017) Sensitivity analysis in keystroke dynamics using Convolutional Neural Networks. In: proc. workshop on information forensics and security
40. Giot R, El-Abed M, Rosenberger C (2009) GREYC Keystroke: a benchmark for keystroke dynamics biometric systems. In: proc. int. conf. on biometrics: theory, applications, and systems
41. Crawford H, Ahmadzadeh E (2017) Authentication on the go: assessing the effect of movement on mobile device keystroke dynamics. In: proc. symp. on usable privacy and security
42. Kim J, Kim H, Kang P (2018) Keystroke dynamics-based user authentication using freely typed text based on user-adaptive feature extraction and novelty detection. Appl Soft Comput 62:1077–1087
43. Murphy , Huang J, Hou D, Schuckers S (2017) Shared dataset on natural human-computer interaction to support continuous authentication research. In: proc. int. joint conf. on biometrics
44. Monaco JV, Tappert CC (2018) The partially observable hidden markov model and its application to keystroke dynamics. Pattern Recogn 76:449–462
45. Bakelman N, Monaco J.V, Cha S.-H, Tappert C.C (2013) Keystroke biometric studies on password and numeric keypad input. In: proc. European intelligence and security informatics conf
46. Coakley M.J, Monaco J.V, Tappert C.C (2016) Keystroke biometric studies with short numeric input on smartphones. In: proc. int. conf. on biometrics theory, applications and systems
47. Monaco J.V, Bakelman N, Cha S.-H, Tappert C.C (2013) Recent advances in the development of a long-text-input keystroke biometric authentication system for arbitrary text input. In: proc. european intelligence and security informatics conf., pp. 60–66
48. Villani M, Tappert C, Ngo G, Simone J, Fort H.S, Cha S.-H (2006) Keystroke biometric recognition studies on long-text input under ideal and application-oriented conditions. In: proc. conf. on computer vision andpattern recognition workshop
49. Cilia D, Inguanez F (2018) Multi-model authentication using keystroke dynamics for smartphones. In: proc. int. conf. on consumer electronics
50. Lu X, Zhang S, Hui P, Lio P (2020) Continuous authentication by free-text keystroke based on CNN and RNN. Comput Secur 96:101861
51. Sun Y, Ceker H, Upadhyaya S (2016) Shared keystroke dataset for continuous authentication. In: proc. int. workshop on information forensics and security
52. Kim J, Kang P (2020) Freely typed keystroke dynamics-based user authentication for mobile devices based on heterogeneous features. Pattern Recogn 108:107556

53. Ayotte B, Banavar M, Hou D, Schuckers S (2020) Fast free-text authentication via instance-based keystroke dynamics. IEEE Trans Biom Behav Identity Sci 2(4):377–387
54. Dhakal V, Feit A.M, Kristensson P.O, Oulasvirta A (2018) Observations on typing from 136 million keystrokes. In: proc. chi conf. on human factors in computing systems
55. El-Kenawy E-SM, Mirjalili S, Abdelhamid AA, Ibrahim A, Khodadadi N, Eid MM (2022) Meta-heuristic optimization and keystroke dynamics for authentication of smartphone users. Mathematics 10(16):2912
56. El-Abed M, Dafer M, Khayat R.E (2014) RHU Keystroke: a mobile-based benchmark for keystroke dynamics systems. In: proc. int. carnahan conf. on security technology, pp. 1–4
57. Al-Obaidi N.M, Al-Jarrah M.M (2016) Statistical median-based classifier model for keystroke dynamics on mobile devices. In: proc. int. conf. on digital information processing and communications, pp. 186–191
58. Stylios I, Skalkos A, Kokolakis S, Karyda M (2022) BioPrivacy: Development of a keystroke dynamics continuous authentication system. In: proc. computer security. ESORICS 2021 Int. workshops
59. Li J, Chang H.-C, Stamp M (2022) Free-text keystroke dynamics for user authentication. Artif Intell Cybersecur, 357–380
60. Tolosana R, Vera-Rodriguez R, Fierrez J, Ortega-Garcia J (2018) Exploring recurrent neural networks for on-line handwritten signature biometrics. IEEE Access 6:5128–5138
61. Tolosana R, Vera-Rodriguez R, Fierrez J, Ortega-Garcia J (2020) BioTouchPass2: touchscreen password biometrics using time-aligned recurrent neural networks. IEEE Trans Inf Forensics Secur 5:2616–2628
62. Neacsu T, Poncu T, Ruseti S, Dascalu M (2023) Doublestrokenet: bigram-level keystroke authentication. Electronics 12(20):4309
63. Wu H, Xu J, Wang J, Long M (2021) Autoformer: Decomposition Transformers with auto-correlation for long-term series forecasting. In: Proc. advances in neural information processing systems
64. Hutchins D, Schlag I, Wu Y, Dyer E, Neyshabur B (2022) Block-recurrent Transformers. In: Proc. advances in neural information processing systems
65. Li B, Cui W, Wang W, Zhang L, Chen Z, Wu M (2021) Two-stream convolution augmented transformer for human activity recognition. In: Proc. AAAI conf. on artificial intelligence
66. Delgado-Santos P, Tolosana R, Guest R, Deravi F, Vera-Rodriguez R (2023) Exploring Transformers for behavioural biometrics: a case study in gait recognition. Pattern Recogn 143:109798
67. Zhou C, Li Q, Li C, Yu J, Liu Y, Wang G, Zhang K, Ji C, Yan Q, He L, Peng H, Li J, Wu J, Liu Z, Xie P, Xiong C, Pei J, Yu P.S, Sun L (2023) A comprehensive survey on pretrained foundation models: a history from BERT to ChatGPT. arXiv:2302.09419
68. Han K, Wang Y, Chen H, Chen X, Guo J, Liu Z, Tang Y, Xiao A, Xu C, Xu Y, Yang Z, Zhang Y, Tao D (2023) A survey on vision transformer. IEEE Trans Pattern Anal Mach Intell 45(1):87–110
69. Wen Q, Zhou T, Zhang C, Chen W, Ma Z, Yan J, Sun L (2022) Transformers in time series: a survey. arxiv:2202.07125
70. Vertanen K, Kristensson P.O (2011) 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
71. Graff D, Cieri C (2003) English Gigaword LDC2003T05. Linguistic Data Consortium, Philadelphia
72. Monaco JV, Tappert CC (2018) The partially observable hidden markov model and its application to keystroke dynamics. Pattern Recognit 76:449–462
73. Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:11
74. Tolosana R, Vera-Rodriguez R et al (2022) SVC-onGoing: signature verification competition. Pattern Recogn 127:108609
75. Stragapede G, Vera-Rodriguez R, Tolosana R, Morales A, Fierrez J, Ortega-Garcia J, Rasnayaka S, Seneviratne S, Dissanayake V, Liebers J, Islam A, Belhaouari S.B, Ahmad S, Jabin S (2022) IJCB 2022 Mobile behavioral biometrics competition (MobileB2C). In: Proc. Int. joint conf. on biometrics
76. Melzi P, Tolosana R. Vera-Rodriguez, R, Kim M, Rathgeb C, Liu X, DeAndres-Tame I, Morales A, Fierrez J, Ortega-Garcia J, et al (2024) FRCSyn-onGoing: benchmarking and comprehensive evaluation of real and synthetic data to improve face recognition systems. Inf Fusion 107:102322
77. Schroff F, Kalenichenko D, Philbin J (2015) FaceNet: A unified embedding for face recognition and clustering. In: Proc. Conf. on computer vision and pattern recognition
78. Deandres-Tame I, Tolosana R, Vera-Rodriguez R, Morales A, Fierrez J, Ortega-Garcia J (2024) How good is chatgpt at face biometrics? a first look into recognition, soft biometrics, and explainability. IEEE Access 12:34390–34401
79. Melzi P, Rathgeb C, Tolosana R, Vera-Rodriguez R, Busch C (2022) An overview of privacy-enhancing technologies in biometric recognition. arXiv:2206.10465