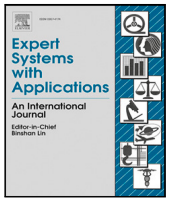
Expert Systems With Applications 237 (2024) 121537

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



SwipeFormer: Transformers for mobile touchscreen biometrics

Paula Delgado-Santos [a,b,*], Ruben Tolosana [b], Richard Guest [a], Parker Lamb [c], Andrei Khmelnitsky [c], Colm Coughlan [c], Julian Fierrez [b]

[a] School of Engineering, University of Kent, United Kingdom
[b] Biometrics and Data Pattern Analytics Lab, Universidad Autonoma de Madrid, Spain
[c] Callsign Inc., United Kingdom

ARTICLE INFO

Keywords:
Behavioural biometrics
Touchscreen
Swipe verification
Transformers
Deep learning
Mobile devices

ABSTRACT

The growing number of mobile devices over the past few years brings a large amount of personal information, which needs to be properly protected. As a result, several mobile authentication methods have been developed. In particular, behavioural biometrics has become one of the most relevant methods due to its ability to extract the uniqueness of each subject in a secure, non-intrusive, and continuous way. This article presents SwipeFormer, a novel Transformer-based system for mobile subject authentication by means of swipe gestures in an unconstrained scenario (i.e., subjects could use their personal devices freely, without restrictions on the direction of swipe gestures or the position of the device). Our proposed system contains two modules: (i) a Transformer-based feature extractor, and (ii) a similarity computation module. Mobile data from the touchscreen and different background sensors (accelerometer and gyroscope) have been studied, including in the analysis both Android and iOS operating systems. A complete analysis of SwipeFormer is carried out using an in-house large-scale database acquired in unconstrained scenarios. In these operational conditions, SwipeFormer achieves Equal Error Rate (EER) values of 6.6% and 3.6% on Android and iOS respectively, outperforming the state of the art. In addition, we evaluate SwipeFormer on the popular publicly available databases Frank DB and HuMIdb, achieving EER values of 11.0% and 5.0% respectively, outperforming previous approaches under the same experimental setup.

1. Introduction

The increasing number of mobile devices in recent years has made them part of our daily lives. As a consequence, mobile devices have become datahubs, including sensitive data as personal or financial details (Delgado-Santos, Stragapede, Tolosana, Guest, Deravi, & Vera-Rodriguez, 2022). Therefore, the security and protection of them through robust and subject-friendly methods are of vital importance (Melzi, Rathgeb, Tolosana, Vera-Rodriguez, & Busch, 2022). Motivated by this fact, biometrics have become one of the most popular authentication methods in mobile devices. In particular, behavioural biometrics, such as gait (Delgado-Santos, Tolosana, Guest, Deravi, & Vera-Rodriguez, 2023; Delgado-Santos, Tolosana, et al., 2022), keystroke dynamics (Stragapede, Vera-Rodriguez, Tolosana, Morales, Acien, & Le Lan, 2023; Stragapede et al., 2022), touchscreen gestures (Fierrez, Pozo, Martinez-Diaz, Galbally, & Morales, 2018; Tolosana, Vera-Rodriguez, Fierrez, & Ortega-Garcia, 2020), and on-line handwritten signature (Tolosana, Vera-Rodriguez, Gonzalez-Garcia, et al., 2022), among others, have shown remarkable results in operational conditions. These authentication techniques provide passive and continuous protection without the need for the subject to perform any specific activity (Patel, Chellappa, Chandra, & Barbello, 2016).

Among the different behavioural biometric traits, keystroke dynamics or mouse dynamics have been traditionally more accurate than touchscreen biometrics. This is because touchscreen gestures, such as swipe or tap, usually consist of simple and short interactions, with a considerable intra-subject variability, making the authentication task more challenging (Stragapede et al., 2022). Nevertheless, the applicability of robust authentication methods based on touchscreen biometrics is crucial to further improve the security of mobile devices in a continuous way, as most of the time our interaction is based on simple tap and swipe gestures (Frank, Biedert, Ma, Martinovic, & Song, 2012).

This article presents SwipeFormer, a novel mobile touchscreen verification system based on Transformers that overcomes some of the drawbacks presented in the literature. Transformers are Deep Learning

* Corresponding author at: School of Engineering, University of Kent, United Kingdom.
E-mail addresses: p.delgado-de-santos@kent.ac.uk (P. Delgado-Santos), ruben.tolosana@uam.es (R. Tolosana), r.m.guest@kent.ac.uk (R. Guest), parker.lamb@uam.es (P. Lamb), andrei.khmelnitsky@uam.es (A. Khmelnitsky), colm.cough@uam.es (C. Coughlan), julian.fierrez@uam.es (J. Fierrez).

https://doi.org/10.1016/j.eswa.2023.121537
Received 15 March 2023; Received in revised form 4 July 2023; Accepted 8 September 2023
Available online 16 September 2023

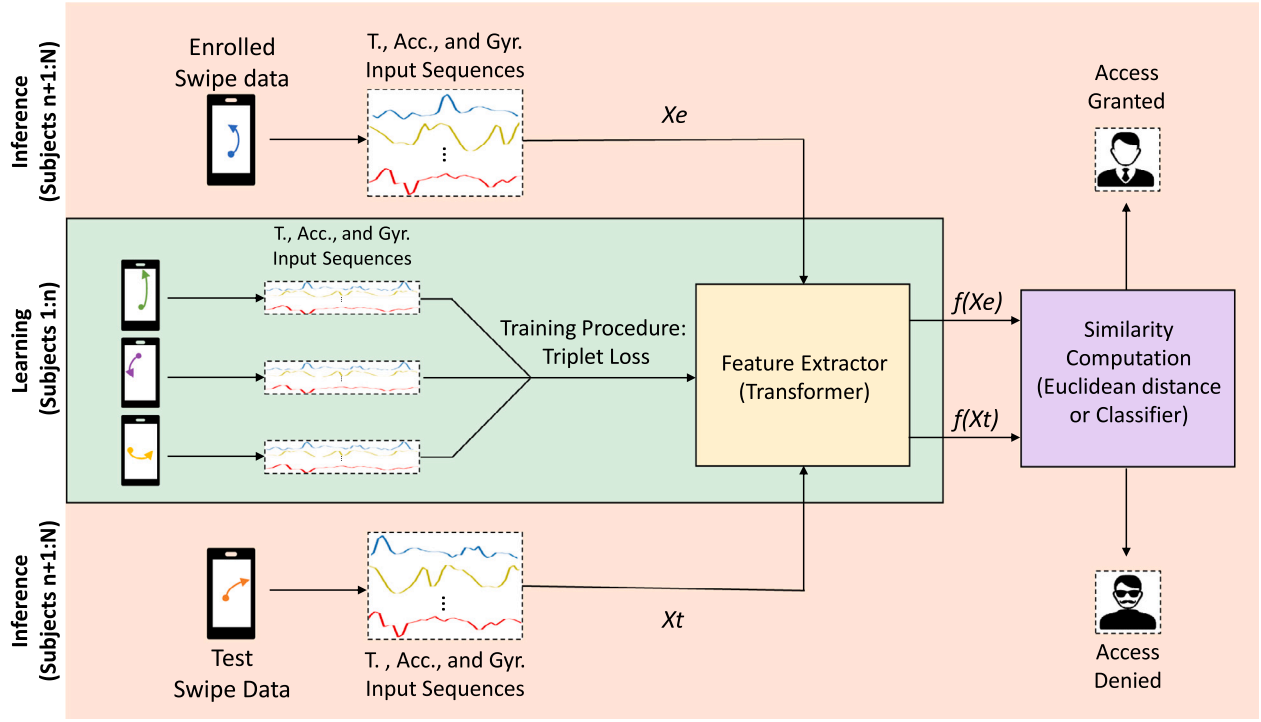


Fig. 1. Graphical representation of SwipeFormer, the proposed mobile touchscreen biometric verification system based on Transformers. N — total number of subjects; X_e — Enrolled swipe sequences; X_t — Test swipe sequences; $f(X_e)$ — Enrolled feature vector; $f(X_t)$ — Test feature vector; T. — Touch; Acc. — Accelerometer; Gyr. — Gyroscope.

(DL) models with an encoder–decoder architecture that have recently achieved impressive results in many fields (e.g., machine translation, computer vision, time series prediction, etc.) due to their extensive modelling skills (Tay, Dehghani, Bahri, & Metzler, 2022). The main advantages of these architectures compared to Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are: (i) all sequences are processed in parallel being feed-forward models; (ii) the self-attention mechanism is implemented on long distance sequences; (iii) more effective training is performed by processing all samples in one batch; and (iv) the entire sequence is taken care of at once, without compressing the previously seen information (Vaswani et al., 2017).

In particular, our proposed touchscreen verification system processes unconstrained (a.k.a. in-the-wild) swipe gestures in a free-direction environment (in contrast to popular touchscreen biometric systems that only consider swipe gestures in a specific direction, i.e., horizontal and vertical), considering therefore more challenging and universal scenarios. Fig. 1 provides a graphical representation of SwipeFormer, comprising both learning and inference stages. First, in the learning stage, the feature extractor module based on a novel Transformer architecture is trained with the development data acquired from the touchscreen and the background sensors of the mobile device. Subsequently, the final evaluation dataset is tested using the similarity computation module, which provides a final score comparison (inference stage).

The main contributions of this article are:

- An in-depth analysis of state-of-the-art swipe verification approaches in mobile scenarios, detailing key public databases and results.
- The proposal of SwipeFormer, a novel touchscreen biometric verification system based on Transformers. To the best of our knowledge, this is the first study that explores the potential of Transformers for mobile touchscreen biometrics. Fig. 2 provides a graphical representation of the feature extractor based on a novel Transformer architecture. Subsequently, for the final similarity computation, different approaches are analysed, i.e., Euclidean distance, Shrunk Covariance, Kernel Density Estimation (KDE), Gaussian Mixture Model (GMM), One-C lass SVM (OC-SVM), and Binary SVM (B-SVM).
- An exhaustive experimental framework is carried out using an in-house database collected in real operational conditions (in-the-wild). To the best of our knowledge, this is the first study that analyses unconstrained touchscreen gestures, achieving promising results. In addition, we show how the different data sources (i.e., touchscreen and background sensors) contribute to the system performance and the differences among the two most popular operating systems (i.e., Android and iOS). Under this challenging scenario, SwipeFormer is able to achieve impressive Equal Error Rate (EER) values of 6.6% and 3.6% on Android and iOS, respectively, showing that the proposed model is more robust in comparison with recent approaches.
- A validation of the proposed SwipeFormer using the popular publicly available databases collected under constrained conditions: Frank DB (Frank et al., 2012) and HuMIdb (Acien, Morales, Fierrez, Vera-Rodriguez, & Delgado-Mohatar, 2021). SwipeFormer achieves EER values of 11.0% and 5.0% on Frank DB and HuMIdb, respectively, outperforming previous state-of-the-art approaches.
- We make our experimental framework available to the research community in order to advance mobile touchscreen research.[1]

The remainder of the article is organised as follows: Section 2 summarises previous works on touchscreen swipe verification on mobile devices. Section 3 describes the architecture of SwipeFormer, including the Transformer-based feature extractor module and the different similarity computation approaches. Then, in Section 4 the main characteristics of the different databases are included, while in Section 5 the experimental setups are described in detail. Section 6 contains the experimental results of SwipeFormer and comparison with the state of

[1] https://github.com/BiDAlab/SwipeFormer

the art. Finally, the conclusions and future research lines are included in Section 7.

2. Related work

Authentication based on touchscreen biometrics recognises a subject through touch gestures performed on a mobile device screen. Swipe gestures are the most common tasks in touchscreen verification (Frank et al., 2012). Table 1 provides a chronological overview of the main touchscreen verification systems in the literature based on swipe gestures, together with their key aspects. One of the main obstacles in this area, apart from the difficulty of the task itself, is the lack of publicly available databases, as each study usually collects its own data (Lamb, Millar, & Fuentes, 2020). In addition, another problem, is the heterogeneity of the settings in each study, making a fair comparison very difficult. We analyse next the key aspects of the area.

Initially, two authentication modalities can be distinguished in this field: continuous and non-continuous. In the first one, continuous authentication, a subject is verified for a period of time while performing gestures on the touchscreen. One of the first studies in the field was (Frank et al., 2012), presenting the public Frank Database with touchscreen data from 4 different android devices and a total of 41 subjects. The authors proposed a system based on the extraction of 30 handcrafted features and One-Class Support Vector Machine (OC-SVM) classifier, achieving performances between 0.00% EER and 4.00% EER with up to 11 swipes per subject. Furthermore, a subject can also be identified in a non-continuous way, where data are collected beforehand and authentication is performed afterwards (Serwadda et al., 2013). In that study the authors also considered touchscreen data, obtaining performances between 10.50% and 17.20% EER with 28 handcrafted features and Logistic Regression.

In addition to the authentication method, the scenario in which the data are acquired is also crucial. Mainly we can distinguish two groups, constrained and unconstrained (a.k.a. in-the-wild) scenarios. In the constrained scenario, the subjects perform a task where data are analysed in a restricted way, i.e., only accepting in one direction the gestures (vertical or horizontal) and/or position of the device (portrait/landscape) (Frank et al., 2012; Serwadda et al., 2013; Xu et al., 2014). On the contrary, in the unconstrained scenario, the data are collected while the subjects use the device freely (Bo et al., 2014; Feng et al., 2014).

In the past few years, the research community has focused on the manual extraction of an optimal set of features from the touchscreen, and their subsequent input into a Machine Learning (ML) model used as a classifier for the verification task. The most popular classifier was One-Class Support Vector Machine (OC-SVM) (Frank et al., 2012). The authors in Saravanan et al. (2014) were able to achieve 97.90% and 96.80% accuracies using a Google Nexus 4 Phone and a Google Nexus 7 Tablet, considering a constrained scenario. Applying the same classifier, in Lu and Liu (2015) the authors achieved 0.03% False Acceptance Rate (FAR) and 0.05% False Rejection Rate (FRR) with a private database. Furthermore, the public HMOG database containing data from the touchscreen and background motion sensors (accelerometer, gyroscope and magnetometer) was presented in Sitová et al. (2015). The authors achieved 8.50% EER using the OC-SVM classifier. In addition, Logistic Regression (Serwadda et al., 2013), Dynamic Time Warping (DTW) (Feng et al., 2014), k-Nearest Neighbours (k-NN) (Antal et al., 2015; Feng et al., 2014) or Random Forest (RF) (Kumar et al., 2016; Mahbub et al., 2016; Saravanan et al., 2014; Shen et al., 2015; Syed et al., 2019) were also broadly used. Another classifier that has been widely used is Binary Support Vector Machine (B-SVM) introduced in Xu et al. (2014). The difference with the previous classifiers is that it needs to be trained using both genuine and impostor data, unlike the previous classifiers which only genuine data are considered. The authors obtained EER values lower than 1% over a private database acquired using only one device and under the continuous authentication scenario.

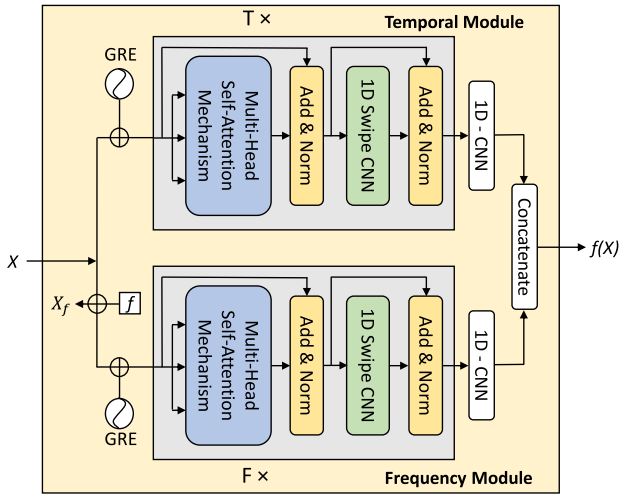


Fig. 2. Graphical representation of the Transformer-based Feature Extractor. X — Input swipe sequence; X_f — Input swipe sequence in the frequency domain; $f(X)$ — Feature vector; GRE — Gaussian Range Encoding; f — Frequency transformation; T ×, F × — Number of layers of each type; 1D-CNN — One Dimension Convolutional Neural Network.

Due to the improvements presented by B-SVM, this classifier has been applied by many studies (Fierrez, Pozo, et al., 2018; Incel et al., 2021; Sharma & Enbody, 2017; Wang et al., 2017). Using each study their own touchscreen data and experimental protocol, the authors achieved 7.00% EER, 80.0% Area Under Curve (AUC) and 2.60% EER, respectively. Moreover, studies based on ML demonstrate how adding extra features from the background sensors of the device to the original touchscreen features improves the performance (Acien, Morales, Vera-Rodriguez, Fierrez, & Tolosana, 2019; Bo et al., 2014; Sitová et al., 2015). For example, in Siirtola et al. (2018) the authors achieved on the HMOG database a 7.00% EER when combining touchscreen and accelerometer data.

In recent years, advancements in Deep Learning (DL) techniques have led to the utilisation of feed-forward Artificial Neural Networks (ANN) as classifiers. Notably, in a study conducted in Zaliva et al. (2015), a private touchscreen database was used in a constrained scenario, achieving an impressive 99.96% F1-Score using 70% of the data to train. In this work, the authors included two hidden layers, consisting of 50–75 and 30 neurons, respectively. The output layer of the network was equipped with a logistic sigmoid activation function. To preprocess the data and enhance the performance of the classifier, Principal Component Analysis (PCA) was applied, reducing the data's dimensionality. Furthermore, the authors in Meng, Wang, et al. (2018) achieved notable results in an unconstrained scenario using a private touchscreen database. Their approach yielded an impressive Average Error Rate (AER) of 2.40%. The proposed model in their work combined Particle Swarm Optimisation (PSO) with an RBFN (Radial Basis Function Network) classifier, which consisted of three layers: an input layer, a hidden layer, and an output layer. Notably, in the hidden layer, each unit adopted a radial activation function, contributing to the model's effective representation and classification capabilities. These findings highlight the potential of utilising PSO and RBFN-based classifiers in touch-based interaction systems, yielding promising results in unconstrained scenarios. In addition, LSTM architectures have shown to be well-suited for the task. In Mao et al. (2022) the authors proposed a 1D-CNN-BiLSTM model that combines the strengths of CNNs and bidirectional LSTMs. The model includes a single convolutional layer with ReLU activation to extract relevant features from the input data. A bidirectional LSTM layer is then employed to capture contextual information in both directions. The model was trained and evaluated using 10-fold cross-validation, ensuring robustness and generalisation.

Table 1
Summary of state-of-the-art approaches presented in the literature for mobile touchscreen biometric verification based on swipe gestures. CA — Continuous Authentication; C — Constrained; U-Unconstrained; T. — Touch, Acc. — Accelerometer; Gyr. — Gyroscope; Mag. — Magnetometer; x — x axis; y — axis; p — pressure; t — timestamp; RNN (LSTM) — Recurrent Neural Network (Long Short-Term Memory); CNN — Convolutional Neural Network; BiLSTM — Bidirectional LSTM; OC-SVM — One-Class Support Vector Machine; B-SVM — Binary SVM; DTW — Dynamic Time Warping; k-NN — k Nearest Neighbours; RF — Random Forest; ANN — Artificial Neural Network; IF — Isolation Forest; EM — Expectation Maximisation Clustering; GMM — Gaussian Mixture Model; Eucl. Dist. — Euclidean Distance; Shrunk Cov. — Shrunk Covariance; KDE — Kernel Density Estimation; EER — Equal Error Rate; FAR — False Acceptance Rate; FRR — False Rejection Rate; AUC — Area Under Curve; AER — Average Error Rate.

Study	Database (Public)	CA	Scenario (C/U)	Device	N. of subjects	Features	Dimension feature vector	Sessions	System		Authentication Data/Subject	Best performance [%]
									Feature Extractor	Classifier/ Distance		
Frank et al. (2012)	✓	✓	C	HTC Droid Inc. Google Nexus One Google Nexus S Samsung Galaxy S	41	T. (x, y, p, t, area)	30	2 (≥ 1 week)	Handcrafted	OC-SVM	11 swipes	0.0–4.0 (EER)
Serwadda, Phoha, and Wang (2013)	✓	✗	C	Google Nexus S	191	T. (x, y, p, t, area)	28	2	Handcrafted	Logistic Regression	80 swipes	10.5–17.2 (EER)
Xu, Zhou, and Lyu (2014)	✗	✓	C	Samsung Galaxy S2	28	T. (x, y, p, t, area)	37	6	Handcrafted	B-SVM	5 swipes (cross-validation)	< 1 (EER)
Feng, Yang, Yan, Tapia, and Shi (2014)	✗	✓	U	Samsung Galaxy S3 Samsung Galaxy S4 Google Nexus 4	23 (+100 test)	T. (x, y, t)	6	3	Handcrafted	DTW + k-NN	200 swipes	90.0 (Accuracy)
Bo et al. (2014)	✗	✓	U	HTC EVO 3D Samsung Galaxy S3	10 (+ 90 test)	T (x, y, p, t), Acc., Gyr.	5	1 day data	Handcrafted	OC-SVM	3 swipes	1 swipe: 23.0 (FAR) 12 swipes: 0.0 (FAR)
Saravanan, Clarke, Chau, and Zha (2014)	✗	✓	C	Google Nexus 4 Phone Google Nexus 7 Tablet	10 (+ 10 test)	T. (x, y, p, t)	4	–	Handcrafted	OC-SVM + RF	–	Phone: 97.9 (Accuracy) Tablet: 96.8 (Accuracy)
Zaliva, Melicher, Saha, and Zhang (2015)	✗	✗	C	Samsung Galaxy S4	14	T. (x, y, z, area)	24	15 min	Handcrafted	ANN	5 swipes	99.96 (F1-Score)
Lu and Liu (2015)	✗	✓	C	Personal	60	T. (x, y, p, t, area)	14	1 month	Handcrafted	OC-SVM	100 swipes	0.03 (FAR) 0.05 (FRR)
Zhang, Patel, Fathy, and Chellappa (2015)	✗	✗	C	iPhone 5S	50	T. (x, y, p, t, area)	27	3	Handcrafted	KDTGR	random 80 swipes	11 swipes: 2.91 (EER) Frank DB: 3.10 (EER) Serwadda DB: 1.73 (EER)
Antal, Bokor, and Szabó (2015)	✓	✓	C	4 Android devices	71	T. (x, y, p, t, area)	15	1 month	Handcrafted	k-NN	100 swipes	1 swipe: 65.0 (Accuracy) 20 swipes: 100.0 (Accuracy)
Shen, Zhang, Guan, and Maxion (2015)	✗	✓	C	Samsung Galaxy N7100 Samsung Galaxy N9002 Huawei Ascend Mate	71	T. (x, y, p, t, area)	22–27	3	Handcrafted	RF	640 swipes	11 swipes: 1.8 (EER)
Sitová et al. (2015)	✓	✓	C	Samsung Galaxy S4	100	T. (x, y, p, t, area) Acc. (x, y, z) Gyr. (x, y, z) Mag. (x, y, z)	71	4	Handcrafted	OC-SVM	≥ 80 swipes (2 sessions)	8.5 (EER)
Mahbub, Sarkar, Patel, and Chellappa (2016)	✓	✓	U	Google Nexus 5	48	T. (x, y, p, t)	24	2 months	Handcrafted	RF	70% swipes	6 swipes: 22.1 (EER)
Sharma and Enbody (2017)	✗	✓	C	Google Nexus 7	42	T. (x, y, p, t, area)	7	40 min	Handcrafted	B-SVM	random 80 swipes	7.0 (EER)
Wang, Yu, Mengshoel, and Tague (2017)	✗	✓	U	Google Nexus 2 Google Nexus 4 Google Nexus 7	20	T. (x, y, p, t, area)	59	4 (1 per device)	Handcrafted	B-SVM	75% swipes	80.0 (AUC)
Kumar, Phoha, and Serwadda (2016)	✗	✓	U	Personal	28	T. (x, y, p, t, area)	5	4-7 days	Handcrafted	RF	50% swipes	99.33 (Accuracy)
Filippov, Iuzbashev, and Kurnev (2018)	–	✓	C	–	20	T. (x, y, t, area)	10	1 month	Handcrafted	IF	2000 swipes	7.5 (FAR) 6.4 (FRR)
Siirtola, Komulainen, and Kellokumpu (2018)	–	✓	C	Samsung Galaxy S4	100	T. (x, y, p, t, area) Acc. (x, y, z)	211	4	Handcrafted	EM	50% swipes	HMOG DB (Read and walk): 7.0 (EER)

(*continued on next page*)

Table 1 (*continued*).

Study	Database (Public)	CA	Scenario (C/U)	Device	N. of subjects	Features	Dimension feature vector	Sessions	System		Authentication Data/Subject	Best performance [%]
									Feature Extractor	Classifier/ Distance		
Fierrez, Pozo, et al. (2018)	✓	✗	C	–	Frank DB: 41 Serwadda DB: 191 Antal DB: 71 UMDAA-02: 48	All DB: T. (x, y, p, t, area)	28	Frank DB: 2 Serwadda DB: 2 Antal DB: 71 UMDAA-02: 2 months	Handcrafted	B-SVM + GMM	40 swipes	Frank DB intra-session: 3.1 (EER) Frank DB inter-session: 8.1 (EER) Serwadda DB intra-session: 3.3 (EER) Serwadda DB inter-session: 10.7 (EER) Antal DB intra-session: 2.6 (EER) UMDAA-02 intra-session: 3.6 (EER)
Meng, Wang, Wong, Wen, and Xiang (2018)	✗	✗	U	Google Nexus 1	48	T. (x, y, t)	21	20	Handcrafted	ANN	60% sessions	2.4 (AER)
Meng, Li, and Wong (2018)	✗	✗	U	Google Nexus 1	60	T. (x, y, p, t, area)	9	30	Handcrafted	SVM	67% sessions	4.7 (AER)
Syed, Helmick, Banerjee, and Cukic (2019)	✗	✗	C	Samsung Tab 210" Samsung Tab 27" Samsung S3 HTC EVO 4G LTE	31	T. (x, y, p, t, area)	18	8 (2-3 weeks)	Handcrafted	RF	50% swipes	3.80 (EER)
Acien, Morales, Vera-Rodriguez, and Fierrez (2020) HuMIdb	✓	✗	C	Personal (Android)	600	T. (x, y, p)	64	≤ 5 ($\geq$ 1 day)	Handcrafted + LSTM	Eucl. Dist.	70% swipes	13.00 (EER)
Incel et al. (2021)	✗	✓	C	Samsung Galaxy S9 Xiaomi Mi8	45	T. (x, y, p) Acc. (x, y, z) Gyr. (x, y, z) Mag. (x, y, z)	54	≤ 3 (same day)	Handcrafted	B-SVM	80% (5-fold cross-validation)	3.50 (EER)
Mao et al. (2022)	✓	✓	C	Android	100	Acc. (x, y, z) Gyr. (x, y, z) Mag. (x, y, z)	57	≤ 24	Handcrafted	CNN-BiLSTM	90% (10-fold cross-validation)	0.53 (EER)
SwipeFormer (2023)	✓	✗	**U**	**Personal**	**Android: 232 iOS: 232**	**T. (x, y, area) Acc. (x, y, z Gyr. (x, y, z)**	**64**	**2** ($\geq$ **1 week**)	**Transformer**	**Eucl. Dist. Shrunk Cov. KDE GMM OC-SVM B-SVM**	**50% swipes**	**In-House DB: Android - 6.6 (EER) In-House DB: iOS - 3.6 (EER) Frank DB - 11.3 (EER) HuMIdb - 5.6 (EER)**

The results highlight the effectiveness of the 1D-CNN-BiLSTM model in touch-based interaction systems. Lastly, in Acien et al. (2020), a Siamese RNN with two LSTM layers was introduced. The model learns to project embedding vectors to differentiate touch patterns from the same and different subjects. By computing the Euclidean distance between embedding vectors, a performance of 13.00% EER was achieved by training the model with 70% of each subject's swipes. In addition, the authors presented a publicly available database, HuMIdb (Acien et al., 2021).

Finally, for completeness, we include in Table 1 the results achieved by our proposed system, SwipeFormer. It is important to highlight that, unlike previous approaches in the literature that consider intra-session variability in their best-case scenario (Fierrez, Pozo, et al., 2018; Zhang et al., 2015), we follow the inter-session experimental protocol proposed in Fierrez, Pozo, et al. (2018) where enrolment and test samples are from different sessions in time (different days), being a more realistic and challenging scenario for behavioural biometrics. The results achieved demonstrate the potential of recent Transformer architectures for the task of mobile touchscreen swipe verification. In addition, we also analyse for the first time in the literature the real performance of swipe biometrics in operational conditions using Android and iOS devices. Also, we consider an unconstrained scenario with swipe gestures performed freely in terms of the position of the devices (portrait/landscape) and direction of the swipe gestures (vertical/horizontal).

3. Proposed system: SwipeFormer

Fig. 1 provides a general representation of SwipeFormer, our proposed touchscreen verification system for mobile scenarios. First, time sequences from the touchscreen and background sensors (i.e., accelerometer and gyroscope) are captured and introduced as the input of the system (X). Subsequently, SwipeFormer consists of two modules: (i) a feature extractor based on a novel Transformer architecture, trained in the learning stage with a development dataset; and (ii) a similarity computation module, which provides the final similarity scores using an evaluation dataset based on subjects not seen in the learning stage (inference stage). The specific details of each module are described next.

3.1. Feature extractor

Fig. 2 shows a graphical representation of the Transformer-based feature extractor trained in the learning stage, based on a novel architecture. The original Transformer, known as Vanilla Transformer, was introduced by Vaswani et al. (2017) for machine translations. That architecture showed impressive results, paving the way for research in other fields such as time sequences (Delgado-Santos et al., 2023; Stragapede, Delgado-Santos, et al., 2022). Despite this, Transformers have numerous drawbacks that have been addressed in the literature. Some of the amendments presented are reducing complexity, including periodicity-based dependencies, or time-depending encoding (Tay et al., 2022).

To overcome some of these disadvantages in the touchscreen biometric scenario, our proposed Transformer comprises two parallel modules: (i) a Temporal Module, which extracts features in the temporal domain; and (ii) a Frequency Module, which extracts discriminative features in the frequency domain. Although in Zhang et al. (2022) the authors demonstrate that models of attention in various domains

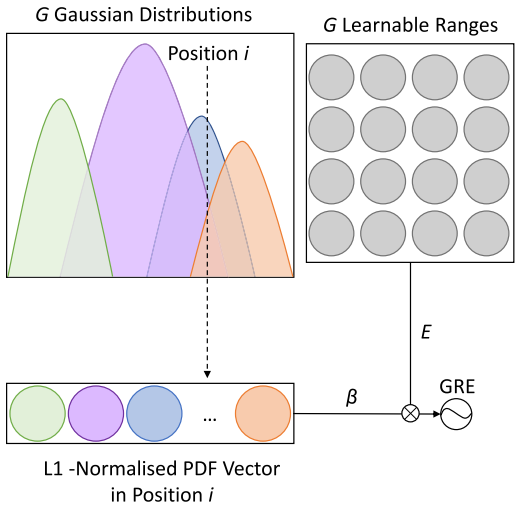


Fig. 3. Graphical representation of the Gaussian range encoding. PDF: Probability Density Function.

(i.e., temporal and frequency) are considered equivalent when exposed to linear conditions, the study also demonstrates the various behaviours exhibited in different domains.

Analysing the Temporal Module first, the input swipe sequence X with L time samples is shaped by a Gaussian Range Encoding (GRE) (See Fig. 3). In line with the idea presented by Li et al. (2021), the GRE is included beforehand to preserve the temporal information. Each position i of the input swipe sequence X is modelled by the Probability Density Functions (PDFs) of G learnable Gaussian distributions. Then, the PDFs are L1-normalised and combined into a vector. After this, the PDFs are computed at G learnable ranges. Finally, the GRE output, X', is the matrix multiplication of the PDFs, β, and the range embeddings E of the input swipe sequence X:

$$X' = X + \beta E \tag{1}$$

Following the GRE, the Temporal Module contains a sequential stack of T layers. Each layer contains two sub-layers: (i) a multi-head self-attention mechanism, and (ii) a one-dimensional multi-scale swipe CNN created specifically for this task. First, the multi-head self-attention mechanism obtains dependencies on the swipe sequence without having a window size limit. This allows all samples from a swipe sequence to be connected to each other. The result of the self-attention mechanism is the weighted summation of the values V in accordance with the dot-product of the queries Q and the matching keys K (Vaswani et al., 2017):

$$\text{Attention}(Q, K, V) = \text{SoftMax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

where d_k is the dimension of the queries Q and keys K, and $\sqrt{d_k}$ is a scaling factor that enables flatter gradients. The output of the sub-layer is the concatenation of applying the self-attention mechanism to H independent heads. As a result, the output has the same dimension as the input sequence, L. Then, the one-dimensional multi-scale swipe CNN comprises three convolutional layers with ReLU activations and different kernel sizes. A batch normalisation and a dropout layer are introduced in between. In addition, each sub-layer is followed by a residual connection and a layer normalisation (Add & Norm in Fig. 2).

Considering the Frequency Module, the input swipe sequence X is represented in the frequency domain by a discrete Fourier transformation X_f. After this, a GRE is included to preserve frequency information. Following an identical architecture to the Temporal Module, the Frequency Module contains the same two sub-layers: (i) a multi-head self-attention mechanism, and (ii) a one-dimensional multi-scale swipe CNN. Each sub-layer is also followed by a residual connection and a layer normalisation (Add & Norm in Fig. 2).

After the Time and Frequency Modules, a one-dimensional convolutional block is included similar to Delgado-Santos et al. (2023). The features extracted by each module are concatenated and fed into a dense layer with sigmoid activation, obtaining the output vector $f(X)$:

$$f(X) = [\text{CNN}(f_t(X)); \text{CNN}(f_f(X))] \tag{3}$$

where $f_t(X)$ and $f_f(X)$ are the extracted features from the Temporal and Frequency Modules respectively.

3.2. Similarity computation

The feature vectors extracted from the enrolled $f(X_e)$ and test $f(X_t)$ swipe sequences are introduced in the similarity computation module to obtain the final similarity score as described in Fig. 1. Six different approaches are considered at inference stage: (i) Euclidean distance, (ii) Shrunk Covariance, (iii) KDE, (iv) GMM, (v) OC-SVM, and (vi) B-SVM.

3.2.1. Euclidean distance

A popular and simple approach widely used in biometrics as it does not require any training. It simple compares the similarity between feature vectors based on the subtraction. Euclidean distance calculates the distance between $f(X_e)$ and $f(X_t)$:

$$d(X_e, X_t) = \sqrt{(f(X_e) - f(X_t))^2} \tag{4}$$

3.2.2. Shrunk covariance

Shrunk Covariance reduces the ratio between the smallest and the largest eigenvalues of the empirical covariance matrix finding the l2-penalised Maximum Likelihood Estimator of the covariance matrix. This matrix is commonly used to model the statistical relationships among the features extracted from biometric samples. The Shrunk covariance is fitted for each subject and tested with samples from the same subject (genuine), and from other subjects in the final evaluation dataset (impostor).

3.2.3. Kernel density estimator (KDE)

This estimator applies kernel smoothing for probability density estimation. KDE is a flexible and powerful tool for analysing and modelling biometric trait distributions, enabling a good understanding of data. Each estimator is trained on data from a single subject and tested on genuine and impostor samples.

3.2.4. Gaussian mixture model (GMM)

The Gaussian Mixture Model shapes the feature vector from a subject into a series of Gaussians in a probabilistic way with Expectation–Maximisation (EM) algorithm. GMM is well-known method in biometrics as it can represent complex, multi-modal distributions, capture intra-class variability, and reduce dimensionality. For the final evaluation, the model is tested with samples from the same subject (genuine), and from other subjects in the final evaluation dataset (impostor).

3.2.5. One-class support vector machine (OC-SVM)

A specific SVM classifier is trained per subject. This configuration may be suitable for many application scenarios as it only considers data from the genuine subject, mapping the data into a high-dimensional feature space where a linear decision boundary can effectively separate the target class from outliers. In the one-class configuration only the enrolled samples of the subject are considered to train the SVM.

Table 2
Summary of the main characteristics of the databases considered in this study together with their experimental setup. T. — Touch, Acc. — Accelerometer, Gyr. — Gyroscope; x — x axis; y — axis; p — pressure.

Database	Subjects	Device	Sessions	Features	Length swipes
In-House	Android: 232 iOS: 232	Free (Android & iOS)	2 (≥ 1 week)	T. (x, y, t, area) Acc. (x, y, z) Gyr. (x, y, z)	Android: 30 iOS: 10
Frank DB (Frank et al., 2012)	41	HTC Droid Inc. Google Nexus One Google Nexus S Samsung Galaxy S	2 (≥ 1 week)	T. (x, y, p, t, area)	50
HuMIdb (Acien et al., 2021)	600	Free (Android)	≤ 5 (≥ 1 day)	T. (x, y, t, p, area) Acc. (x, y, z) Gyr. (x, y, z) ...	100

3.2.6. Binary support vector machine (b-SVM)

In the case of B-SVM, a subject-specific SVM classifier is trained. In contract to OC-SVM, one classifier is trained using both enrolled samples of the subject and also samples of other subjects (from the development dataset) used as impostor. In cases where genuine and impostor samples are available for training, it is a very powerful classifier as it provides strong generalisation, is robust to overfitting and there are few parameters to adjust.

4. Databases description

Three different databases have been considered in the experimental framework of this study. These databases contain touchscreen data extracted from mobile devices while performing swipe gestures. In particular, we consider an in-house collected database together with two public databases widely used in the literature. In each database, different acquisition conditions and mobiles devices are considered (e.g. sampling rate, screen size). The main characteristics of the databases together with the experimental setup in this study are reported in Table 2.

- **In-House Database:** This database comprises 464 subjects. For each swipe gesture acquired, we have the corresponding information of the touchscreen, accelerometer, and gyroscope sensors. The subjects were required to authenticate themselves in real-world, unconstrained (not supervised) settings. This scenario allowed subjects to use their personal devices freely, performing the gestures in their preferred way, location, and timing. In addition, there were no restrictions in terms of the position of the device (portrait or landscape) and the direction of the gestures (vertical or horizontal). As a result, this database considers real operational conditions, unlike previous swipe databases in the field. The data were collected using each subject's personal smartphone (Android and iOS) over a period of one year, with a minimum one-week gap between at least two sessions. To comply with the General Data Protection Regulation (GDPR) from the European Commission, information related to the device specifications (only the operating system), demographics, and statistics were not collected.
- **Frank Database** (Frank et al., 2012)**:** This database contains swipe data from 41 subjects. Touchscreen data were collected from 4 different Android devices (HTC Droid Inc, Google Nexus One, Google Nexus S, and Samsung Galaxy S) while subjects were comparing images and reading text under constrained conditions. All devices are in portrait orientation. At least 2 sessions per subject separated by 1 week were collected.
- **HuMI Database** (Acien et al., 2021)**:** This database is the largest public mobile touchscreen database available in the literature. Data were collected from 600 subjects in a human-mobile interaction using the touchscreen and different background sensors (e.g., linear accelerometer, accelerometer, and gyroscope, among others). Data were collected using an Android app while subjects performed 8 simple tasks (i.e., keystroke, swipe, tap, audio, and draw a number) on their own devices under constrained conditions. The number of acquisition sessions per subject is 5 or fewer, with at least 1 day in between.

Table 3
Hyperparameters configuration.

Temporal Module	Gaussians in GRE (G) = 20 Temporal Layers (T) = 9 Temporal Heads (H) = 20
Frequency Module	Gaussians in GRE (G) = 20 Frequency Layers (F) = 9 Frequency Heads (H) = 20
Temporal + Frequency Modules	Feature Vector Size (S) = 64

5. Experimental setup

5.1. Feature extractor hyperparameters

The best architecture and hyperparameters of the proposed Transformer have been selected using only the development dataset of the in-house database. This selection has been carried out manually, based on trial and error. Table 3 provides an overview of the key hyperparameters of SwipeFormer. Both Temporal and Frequency modules have the same structure, the only difference is the Fast Fourier Transform (FFT) included in the Frequency Module with an output size of $s = L - 1$, where L is the length (number of samples) of each input swipe sequence. The Gaussian range encoding includes $G = 20$ Gaussian distributions. After them, the Temporal Module comprises $T = 9$ layers, and the Frequency Module contains $F = 9$ layers. Each layer includes $H = 10$ heads. Subsequently, in each module the multi-scale swipe CNN comprises 3 convolutional layers with L units each, and kernel sizes 1, 3, and 5, respectively. In addition, the convolutional layers include ReLU activation functions, followed by dropout layers with a rate of 0.1. Finally, 2 convolutional layers with L units each, ReLU activation functions, and kernel sizes of 512 and 256 are included at the end of each module. The final output vector $f(X)$ contains $S = 64$ features as a result of concatenating the output of the modules fed into the dense layer with a sigmoid activation.

5.2. System details

The Transformer-based feature extractor is trained in the learning stage. The triplet loss strategy is employed for the training, including a Euclidean distance with a margin of $\alpha = 1.0$ for each triplet comparison. Each triplet consists of three swipe gestures (containing each swipe the corresponding information related to the touchscreen, accelerometer, and gyroscope sensors): (i) anchor (belonging to an enrolled

Table 4
Comparison of the performance in EER (%) achieved by the proposed SwipeFormer with different similarity computation approaches in our in-house database (Android and iOS devices). Eucl. Dist. — Euclidean Distance; Shrunk Cov. — Shrunk Covariance; T. — Touch; Acc. — Accelerometer; Gyr — Gyroscope.

Method		Databases			
Feature extractor	Similarity computation	Android		iOS	
		T.	T., Acc., Gyr.	T.	T., Acc., Gyr.
SwipeFormer	Eucl. Dist.	12.3	12.1	13.9	13.7
	Shrunk Cov.	13.0	11.9	9.0	11.2
	KDE	7.8	7.5	7.9	8.5
	GMM	10.4	10.8	8.6	8.1
	OC-SVM	8.7	7.6	8.3	9.9
	B-SVM	6.9	**6.6**	5.3	**3.6**
Fierrez, Pozo, et al. (2018)		43.4	43.2	35.1	34.8
Acien et al. (2020)		18.1	17.7	15.3	14.7

subject), (ii) positive (belonging to the same subject considered in the anchor), and (iii) negative (belonging to a different subject). Triplets are randomly formed using the subjects and swipes gestures included in the training dataset, following the guidelines explained before for the anchor, positive, and negative samples. The Adam optimiser with a learning rate of 0.001 is used. Furthermore, a stop condition is included for the training: if the feature extractor does not improve the validation loss for 10 epochs, the training stops.

In the inference stage, the evaluation of SwipeFormer includes different similarity computation approaches. The Shrunk Covariance and KDE are evaluated with the Mahalanobis distance; and GMM with diagonal covariance. In addition, KDE uses a Guassian kernel and a bandwidth of 0.9. OC-SVM and B-SVM contain an *RBF* kernel with $\gamma = 0.5$.

5.3. Experimental protocol

Next, we describe the experimental protocol details considered in the study. The specifications of each database and stage (learning and inference) are included.

5.3.1. Experiment 1: In-house database

The first experiment analyses the performance of SwipeFormer using the in-house database. Data from the touchscreen (x, y, area), accelerometer (x, y, z), and gyroscope (x, y, z) are included. Two experiments are considered, one for Android and one for iOS operating systems, considering in both the same experimental protocol.

In each experiment, the learning stage consists of 190 subjects, 148 of which belong to the train dataset and 42 to the validation dataset. Regarding the inference stage, the 42 remaining unseen subjects are included in the evaluation dataset. Regarding the learning stage, the training dataset contains in total 18,066 triplets and the validation dataset includes 3,778. Each swipe of each subject represents the anchor of a triplet, the positive pair is randomly selected from another swipe of the same subject, and the negative pair from a random swipe of another subject of the training/validation dataset. Regarding the inference stage, for each subject the final verification scores are obtained comparing 5 enrolled swipes from the first session with 10 test swipes from the last session (genuines) and with 10 swipe from other subjects (impostors). Finally, the Android subset contains a sequence length $L = 30$ while the iOS subset $L = 10$.

5.3.2. Experiment 2: Frank and humi databases

To validate the potential of SwipeFormer, the proposed architecture has been evaluated and compared with the literature using two popular publicly available databases: (i) Frank DB (Frank et al., 2012), which is one of the first public databases in the literature, and (ii) HuMIdb (Acien et al., 2021), which is the largest public database in the field, acquired recently in unconstrained scenarios.

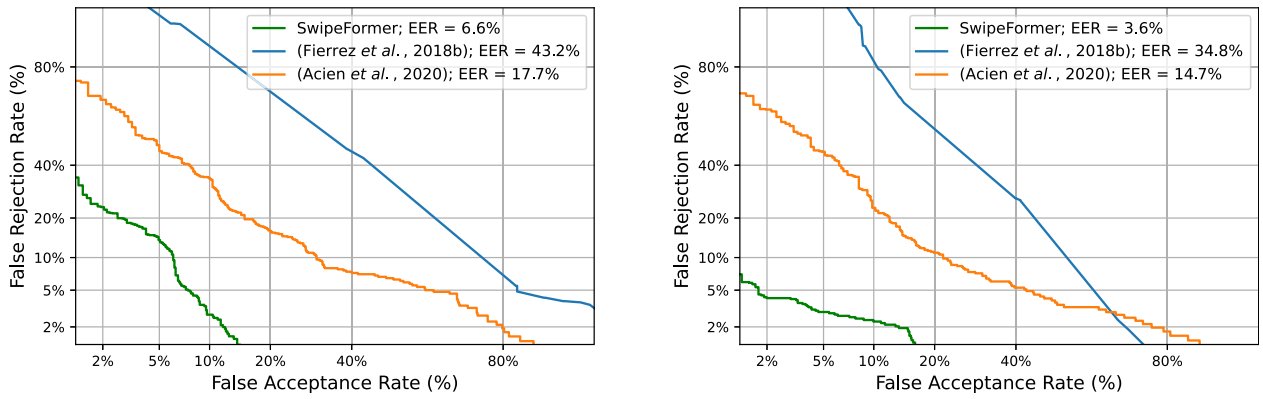
First, in the Frank DB, 33 subjects are included in the development dataset (learning stage) while the remaining 8 subjects are included for the final evaluation (inference stage). Data from the touchscreen (x, y, area) are included. For the learning stage, the training dataset includes 11,694 triplets and the validation dataset contains 3,365. All swipes have $L = 50$ samples. In order to provide a fair comparison with the literature, we follow the same (inter-session) experimental protocol considered in Fierrez, Pozo, et al. (2018) for the inference stage. It is important to highlight that, unlike previous approaches in the literature that consider intra-session variability (Fierrez, Pozo, et al., 2018; Zhang et al., 2015), we follow an inter-session experimental protocol where enrolment and test samples are from different sessions in time (different days), being a more realistic and challenging scenario for behavioural biometrics. For each subject, one session is used for enrolment while the other one is used for test as genuine. Regarding impostor scores, swipes from other random subjects are used as impostor. It is important to highlight that, contrary to previous approaches in the literature such as Fierrez, Pozo, et al. (2018), SwipeFormer considers a more universal scenario, without specifying the model the particular swipe directions (vertical, horizontal, etc.) or position of the device (portrait or landscape). Therefore, we consider in the analysis more challenging scenarios.

In addition, for HuMIdb, we replicate the experimental protocol considered in Acien et al. (2020). In particular, the right-swipe gestures between tasks are included in this study. Data from the touchscreen (x, y, p) are considered. Specifically, 424 subjects are used in the learning stage (24,430 triplets for training and 5,946 triplets for validation), while the remaining 178 unseen subjects are part of the final evaluation (inference stage). All swipes have a length of $L = 100$ samples. Finally, for the inference stage, we consider the first 5 swipes per subject as enrolled swipes, and the last 10 genuine swipes for testing. Furthermore, 10 random swipes from other subjects are included as impostor swipes for testing.

6. Experimental results

6.1. Experiment 1: In-house database

Table 4 shows the results of SwipeFormer in terms of EER (%) for the Android and iOS evaluation datasets and for the different similarity computation configurations considered: Euclidean distance, Shrunk Covariance, KDE, GMM, OC-SVM, and B-SVM. Two different feature configurations are studied: (i) including the touchscreen and (ii) the combination of touchscreen and background sensors (accelerometer and gyroscope). In addition, to provide a better comparison of SwipeFormer with the state of the art, we include in the table the results achieved by recent approaches in the literature, i.e., Acien et al. (2020) and Fierrez, Pozo, et al. (2018).



(a) Android Database Results (b) iOS Database Results

Fig. 4. DET curves and EER (%) achieved by the proposed SwipeFormer and other state-of-the-art approaches in the literature in our in-house database (Android and iOS). T. — Touch, Acc. — Accelerometer, Gyr — Gyroscope.

Analysing the results with Euclidean distance, SwipeFormer achieves EER values of 12.3% for touchscreen and 12.1% for touchscreen and background sensors (relative EER improvement of 1.7%) in the Android configuration; and 13.9% for touchscreen and 13.7% for touchscreen and background sensors (relative EER improvement of 1.4%) in the iOS configuration. These results demonstrate how background sensors on mobile devices can provide additional information (e.g. the uniqueness of the device used and held by the subject).

In addition, we analyse the performance of different similarity computation configurations with the best feature configuration (touchscreen and background sensors) in the Android configuration. Compared with Euclidean distance, other approaches improve this result in terms of ERR with relative improvements of 1.70%, 38.00%, 10.70%, 37.20%, 45.50% for Shrunk Covariance, KDE, GMM, OC-SVM, and B-SVM respectively. These results prove that training each SVM with both genuine (from the enrolled subject) and impostor (from other subjects) swipes, more accurate verification is achieved than with the other configurations.

Furthermore, an analysis of the iOS configuration shows similar behaviour. Our proposed SwipeFormer with the best feature configuration (touchscreen and background sensors) and Euclidean distance achieves an ERR of 13.70%, while other similarity computation approaches relatively improve this result in terms of EER (Shrunk Covariance: 18.20%, KDE: 38.00%, GMM: 41.00%, OC-SVM: 27.70%, B-SVM: 73.70%). These results demonstrate how classifiers such as SVM are able to separate the different classes (genuine and impostor) with a higher margin, achieving better performance.

In addition, as can be seen in Table 4, it is interesting to remark that the iOS configuration reaches in general better performance than the Android configuration, achieving in the best case a relative improvement of 45.50% (3.60% EER vs. 6.60% EER). We hypothesise that this improvement achieved on iOS can be produced due to all devices are from the same company (Apple), using similar high-quality accelerometer and gyroscope sensors, contrary to the Android case that contains very different smartphone models in terms of sensors. The quality and calibration of the sensors, and the device's overall design and hardware integration has been studied in Franček, Jambrošić, Horvat, and Planinec (2023).

Finally, for completeness, Fig. 4 shows the Detection Error Trade-Off (DET) curves of the proposed SwipeFormer and the previous touchscreen biometric systems, (Acien et al., 2020; Fierrez, Pozo, et al., 2018), in the two configurations studied (Android and iOS). The best feature configuration is analysed by combining the touchscreen and the background sensors (accelerometer and gyroscope). Overall, we can see a similar behaviour in both configurations, where SwipeFormer with the B-SVM approach outperforms (Fierrez, Pozo, et al., 2018) with a relative improvement in terms of EER of 158.10% and 89.70% in Android and iOS respectively. A similar trend is observer for Acien et al. (2020) with an EER relative improvement of 62.70% in Android and 75.50% in iOS. These results show how the correct classifier, such as SVM, helps to better adapt the features extracted by the Transformer to each specific subject. This is consistent with related works that have shown subject-adaptation (Fierrez, Morales, Vera-Rodriguez, & Camacho, 2018) to be very useful in behavioural biometrics (Fierrez-Aguilar, Garcia-Romero, Ortega-Garcia, & Gonzalez-Rodriguez, 2005).

6.2. Experiment 2: Frank and humi databases

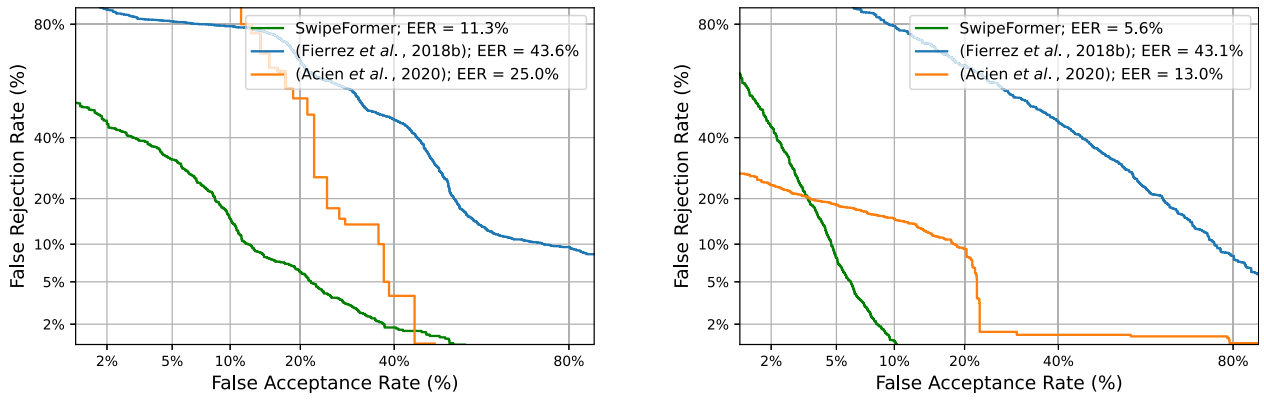
Fig. 5 shows the DET curves and the EER results obtained in Acien et al. (2020) and Fierrez, Pozo, et al. (2018) systems, and the proposed SwipeFormer on the public available databases Frank and HuMIdb. All experiments in each database have the same experimental protocols. Furthermore, B-SVM is considered in the similarity computation module of SwipeFormer.

This experiment proves the robustness of the features extracted by our proposed SwipeFormer. Overall, it can be observed how SwipeFormer outperforms previous state-of-the-art approaches in different databases under the same experimental protocol. In particular, for the Frank database (Frank et al., 2012), SwipeFormer achieves an EER of 11.30% in comparison with the 43.60% EER obtained with Fierrez, Pozo, et al. (2018) and 25.00% with Acien et al. (2020) (relative improvements of 74.80% and 56.00% respectively). In addition, for HuMIdb (Acien et al., 2021), SwipeFormer obtains an EER of 5.60% while Acien et al. (2020) and Fierrez, Pozo, et al. (2018) approaches achieve 43.10% and 13.00% EERs respectively (relative improvements of 88.40% and 61.50%).

The results obtained highlight the significant potential of the proposed Transformer for several reasons. Firstly, the incorporation of GRE allows to introduce in each sample details about its relative position with respect within the sequence, increasing the complexity of the extracted information. Finally, the adoption of a two-stream architecture (Temporal and Frequency Modules) facilitates the extraction of distinct features, obtaining a more complete representation of each sample.

6.3. Deployment on real scenarios

This section analyses the time consumption of the proposed SwipeFormer. It is important to highlight that the training of SwipeFormer, like most DL models, is typically carried out off-line on powerful computers or servers with dedicated GPUs. As indicated in Fig. 1, this corresponds to the training stage. Once the model is trained using large-scale databases, it can be deployed in real-time applications for feature extraction, also known as the inference stage. In addition,



(a) Frank Database Results

(b) HuMI Database Results

Fig. 5. DET curves and EER (%) achieved by the proposed SwipeFormer and other state-of-the-art approaches in the literature, i.e., Acien et al. (2020) and Fierrez, Pozo, et al. (2018). The best configuration of the touchscreen and background sensors (accelerometer and gyroscope) is analysed.

it is important to remark that there are various options for storing and running DL models. For instance, the DL model can be stored on a remote server, which receives input data from the mobile device, calculates the score or prediction, and then sends it back to the mobile device to make a decision based on the obtained result. This approach allows the computational burden to be shifted to the server, leveraging its higher processing capabilities, while the mobile device primarily handles input/output communication and decision-making based on the received scores or predictions.

Therefore, in this section we analyse the time consumption of SwipeFormer in the inference stage, simulating the final application scenario. All experiments are carried out using *PyTorch* with an *Intel Core i7-12700K* processor and an *NVIDIA GeForce RTX 3090* graphics card. For reproducibility reasons, we consider in this analysis the publicly available databases Frank DB and HuMIdb. Regarding the Frank DB, SwipeFormer achieves a significantly reduction in time, only 2.11 ms per comparison (on average), surpassing the models presented by Fierrez, Pozo, et al. (2018) with 567.88 ms and Acien et al. (2020) with 7.66 ms. Similarly, in the case of HuMIdb, SwipeFormer outperforms previous state-of-the-art approaches with a time of 0.34 ms per comparison (on average), compared to 3.39 ms and 7.61 ms achieved by Acien et al. (2020) and Fierrez, Pozo, et al. (2018), respectively. These results demonstrate that SwipeFormer not only improves recognition accuracy but also excels in terms of time efficiency compared to previous approaches.

7. Conclusions and future work

The present study has introduced SwipeFormer, a novel touchscreen verification system based on Transformers. To the best of our knowledge, this is the first attempt to apply Transformers to touchscreen biometrics.

SwipeFormer consists of two modules: (i) a feature extractor based on a Transformer architecture, trained with the development data acquired from the touchscreen and the background sensors of the mobile device in the learning stage; and (ii) a similarity computation module (Euclidean distance, Shrunk Covariance, KDE, GMM, OC-SVM, and B-SVM), which provides a final verification with a evaluation dataset (inference stage). SwipeFormer contains two modules (Temporal and Frequency) with a GRE, multi-head self-attention mechanism, and CNNs.

Two experiments are carried out considering an in-house database collected under unconstrained conditions and two of the most popular public databases in touchscreen biometric verification collected under constrained conditions (Frank DB and HuMIdb). For the publicly available databases, the same experimental protocol proposed in the literature was considered. Regarding the experimental results, SwipeFormer outperforms state-of-the-art systems in all databases, achieving EER of 3.60%, 11.30%, and 5.00% in our in-house database, Frank DB, and HuMIdb respectively.

Future work will explore and analyse Transformers in other behavioural biometric modalities such as handwritten signature (Tolosana, Vera-Rodriguez, et al., 2022) or electrocardiograms (Melzi, Tolosana, & Vera-Rodriguez, 2022). Furthermore, the study of the ageing effect on touchscreen biometrics (Tolosana, Vera-Rodriguez, Fierrez, & Ortega-Garcia, 2019). Finally, future work will try to improve the performance of SwipeFormer considering DL models for the synthesis of data such as Variational Autoencoders (VAEs) or Generative Adversarial Network (GAN). These approaches can significantly improve one- and few-shot learning scenarios as demonstrated in Tolosana et al. (2021).

CRediT authorship contribution statement

Paula Delgado-Santos: Conceptualization, Investigation, Writing – original draft, Writing – review & editing, Visualization, Funding acquisition. **Ruben Tolosana:** Conceptualization, Investigation, Writing – original draft, Writing – review & editing, Visualization, Funding acquisition. **Richard Guest:** Conceptualization, Writing – review & editing, Visualization, Funding acquisition. **Parker Lamb:** Conceptualization, Writing – review & editing. **Andrei Khmelnitsky:** Conceptualization, Writing – review & editing. **Colm Coughlan:** Conceptualization, Writing – review & editing. **Julian Fierrez:** Conceptualization, Writing – review & editing, Visualization, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data already public 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. With support also from projects INTER-ACTION (PID2021-126521OB-I00 MICINN/FEDER) and HumanCAIC (TED2021-131787B-I00 MICINN), and from the Autonomous Community of Madrid (ELLIS Unit Madrid). Part of this work was done while Paula Delgado-Santos was a visiting researcher at Callsign Inc. (UK).

References

Acien, A., Morales, A., Fierrez, J., Vera-Rodriguez, R., & Delgado-Mohatar, O. (2021). BeCAPTCHA: Behavioral bot detection using touchscreen and mobile sensors benchmarked on HuMIdb. *Engineering Applications of Artificial Intelligence*.

Acien, A., Morales, A., Vera-Rodriguez, R., & Fierrez, J. (2020). Smartphone sensors for modeling human-computer interaction: General outlook and research datasets for user authentication. In *Proc. IEEE annual computers, software, and applications conference*.

Acien, A., Morales, A., Vera-Rodriguez, R., Fierrez, J., & Tolosana, R. (2019). Multilock: Mobile active authentication based on multiple biometric and behavioral patterns. In *Proc. international workshop on multimodal understanding and learning for embodied applications*.

Antal, M., Bokor, Z., & Szabó, L. Z. (2015). Information revealed from scrolling interactions on mobile devices. *Pattern Recognition Letters*, *56*, 7–13.

Bo, C., Zhang, L., Jung, T., Han, J., Li, X.-Y., & Wang, Y. (2014). Continuous user identification via touch and movement behavioral biometrics. In *Proc. IEEE international performance computing and communications conference*.

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 Compututing Surveys*.

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 Recognition*.

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 Recognition Letters*, *161*, 30–37.

Feng, T., Yang, J., Yan, Z., Tapia, E. M., & Shi, W. (2014). Tips: Context-aware implicit user identification using touch screen in uncontrolled environments. In *Proc. workshop on mobile computing systems and applications*.

Fierrez, J., Morales, A., Vera-Rodriguez, R., & Camacho, D. (2018). Multiple classifiers in biometrics. Part 2: Trends and challenges. *Information Fusion*, *44*, 103–112.

Fierrez, J., Pozo, A., Martinez-Diaz, M., Galbally, J., & Morales, A. (2018). Benchmarking touchscreen biometrics for mobile authentication. *IEEE Transactions on Information Forensics and Security*, *13*(11), 2720–2733.

Fierrez-Aguilar, J., Garcia-Romero, D., Ortega-Garcia, J., & Gonzalez-Rodriguez, J. (2005). Bayesian adaptation for user-dependent multimodal biometric authentication. *Pattern Recognition*, *38*(8), 1317–1319.

Filippov, A. I., Iuzbashev, A. V., & Kurnev, A. S. (2018). User authentication via touch pattern recognition based on isolation forest. In *Proc. IEEE conference of russian young researchers in electrical and electronic engineering*.

Franček, P., Jambrošić, K., Horvat, M., & Planinec, V. (2023). The performance of inertial measurement unit sensors on various hardware platforms for binaural head-tracking applications. *Sensors*, *23*(2), 872.

Frank, M., Biedert, R., Ma, E., Martinovic, I., & Song, D. (2012). Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, *8*(1), 136–148.

Incel, Ö. D., Günay, S., Akan, Y., Barlas, Y., Basar, O. E., Alptekin, G. I., et al. (2021). Dakota: Sensor and touch screen-based continuous authentication on a mobile banking application. *IEEE Access*, *9*, 38943–38960.

Kumar, R., Phoha, V. V., & Serwadda, A. (2016). Continuous authentication of smartphone users by fusing typing, swiping, and phone movement patterns. In *Proc. IEEE international conference on biometrics theory, applications and systems*.

Lamb, P., Millar, A., & Fuentes, R. (2020). Swipe Dynamics as a Means of Authentication: Results from a Bayesian unsupervised approach. In *Proc. IEEE international joint conference on biometrics* (pp. 1–9).

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 conference on artificial intelligence*.

Lu, L., & Liu, Y. (2015). Safeguard: User reauthentication on smartphones via behavioral biometrics. *IEEE Transactions on Computational Social Systems*, *2*(3), 53–64.

Mahbub, U., Sarkar, S., Patel, V. M., & Chellappa, R. (2016). Active User authentication for smartphones: A challenge data set and benchmark results. In *Proc. IEEE international conference on biometrics theory, applications and systems*.

Mao, R., Ji, H., Cheng, D., Wang, X., Wang, Y., & Sun, D. (2022). Implicit continuous authentication model based on mobile terminal touch behavior. In *Proc. IEEE symposium on computers and communications*.

Melzi, P., Rathgeb, C., Tolosana, R., Vera-Rodriguez, R., & Busch, C. (2022). An overview of privacy-enhancing technologies in biometric recognition. arXiv preprint arXiv:2206.10465.

Melzi, P., Tolosana, R., & Vera-Rodriguez, R. (2022). ECG biometric recognition: Review, system proposal, and benchmark evaluation. arXiv preprint arXiv:2204.03992.

Meng, W., Li, W., & Wong, D. S. (2018). Enhancing touch behavioral authentication via cost-based intelligent mechanism on smartphones. *Multimedia Tools and Applications*, *77*(23), 30167–30185.

Meng, W., Wang, Y., Wong, D. S., Wen, S., & Xiang, Y. (2018). TouchWB: Touch behavioral user authentication based on web browsing on smartphones. *Journal of Network and Computer Applications*, *117*, 1–9.

Patel, V. M., Chellappa, R., Chandra, D., & Barbello, B. (2016). Continuous user authentication on mobile devices: Recent progress and remaining challenges. *IEEE Signal Processing Magazine*, *33*(4), 49–61.

Saravanan, P., Clarke, S., Chau, D. H., & Zha, H. (2014). Latentgesture: Active user authentication through background touch analysis. In *Proc. International Symposium of Chinese CHI*.

Serwadda, A., Phoha, V. V., & Wang, Z. (2013). Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In *Proc. IEEE international conference on biometrics: Theory, applications and systems*.

Sharma, V., & Enbody, R. (2017). User authentication and identification from user interface interactions on touch-enabled devices. In *Proc. ACM conference on security and privacy in wireless and mobile networks*.

Shen, C., Zhang, Y., Guan, X., & Maxion, R. A. (2015). Performance analysis of touch-interaction behavior for active smartphone authentication. *IEEE Transactions on Information Forensics and Security*, *11*(3), 498–513.

Siirtola, P., Komulainen, J., & Kellokumpu, V. (2018). Effect of context in swipe gesture-based continuous authentication on smartphones. In *European symposium on artificial neural networks, computational intelligence and machine learning*.

Sitová, Z., Šeděnka, J., Yang, Q., Peng, G., Zhou, G., Gasti, P., et al. (2015). HMOG: New behavioral biometric features for continuous authentication of smartphone users. *IEEE Transactions on Information Forensics and Security*, *11*(5), 877–892.

Stragapede, G., Delgado-Santos, P., Tolosana, R., Vera-Rodriguez, R., Guest, R., & Morales, A. (2022). TypeFormer: Transformers for mobile keystroke biometrics. arXiv preprint arXiv:2212.13075.

Stragapede, G., Vera-Rodriguez, R., Tolosana, R., & Morales, A. (2023). BehavePassDB: Public database for mobile behavioral biometrics and benchmark evaluation. *Pattern Recognition*, *134*, Article 109089.

Stragapede, G., Vera-Rodriguez, R., Tolosana, R., Morales, A., Acien, A., & Le Lan, G. (2022). Mobile behavioral biometrics for passive authentication. *Pattern Recognition Letters*.

Syed, Z., Helmick, J., Banerjee, S., & Cukic, B. (2019). Touch gesture-based authentication on mobile devices: The effects of user posture, device size, configuration, and inter-session variability. *Journal of Systems and Software*, *149*, 158–173.

Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient Transformers: A survey. *ACM Computing Surveys*.

Tolosana, R., Delgado-Santos, P., andres, P.-U., Vera-Rodriguez, R., Fierrez, J., & Morales, A. (2021). DeepWriteSYN: On-line handwriting synthesis via deep short-term representations. In *Proc. AAAI conference on artificial intelligence*.

Tolosana, R., Vera-Rodriguez, R., Fierrez, J., & Ortega-Garcia, J. (2019). Reducing the template aging effect in on-line signature biometrics. *IET Biometrics*, *8*(6), 422–430.

Tolosana, R., Vera-Rodriguez, R., Fierrez, J., & Ortega-Garcia, J. (2020). BioTouchPass2: Touchscreen password biometrics using time-aligned recurrent neural networks. *IEEE Transactions on Information Forensics and Security*, *15*, 2616–2628.

Tolosana, R., Vera-Rodriguez, R., Gonzalez-Garcia, C., Fierrez, J., Morales, A., Ortega-Garcia, J., et al. (2022). SVC-onGoing: Signature verification competition. *Pattern Recognition*, *127*, Article 108609.

Tolosana, R., Vera-Rodriguez, R., et al. (2022). SVC-onGoing: Signature verification competition. *Pattern Recognition*, *127*, 1–14.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Proc. advances in neural information processing systems*.

Wang, X., Yu, T., Mengshoel, O., & Tague, P. (2017). Towards continuous and passive authentication across mobile devices: An empirical study. In *Proc. ACM conference on security and privacy in wireless and mobile networks*.

Xu, H., Zhou, Y., & Lyu, M. R. (2014). Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *Proc. symposium on usable privacy and security*.

Zaliva, V., Melicher, W., Saha, S., & Zhang, J. (2015). Passive user identification using sequential analysis of proximity information in touchscreen usage patterns. In *Proc. IEEE international conference on mobile computing and ubiquitous networking*.

Zhang, X., Jin, X., Gopalswamy, K., Gupta, G., Park, Y., Shi, X., et al. (2022). First detrend then attend: Rethinking attention for time-series forecasting. arXiv preprint arXiv:2212.08151.

Zhang, H., Patel, V. M., Fathy, M., & Chellappa, R. (2015). Touch gesture-based active user authentication using dictionaries. In *Proc. IEEE winter conference on applications of computer vision*.