Pattern Recognition 180 (2026) 114258



Contents lists available at ScienceDirect

Pattern Recognition

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





ProtoSig: Enhancing training data for offline handwritten signature verification using prototypical signatures

Kecia G. de Moura *, Robert Sabourin, Rafael M.O. Cruz

École de technologie supérieure - Université du Québec, 1100 Notre-Dame St W, Montreal, H3C 1K3, Quebec, Canada

ARTICLE INFO

Keywords:
Offline handwritten signatures
Prototypical signatures
Writer-dependent system
Biometrics
Scalability

ABSTRACT

Offline Handwritten Signature Verification (offline HSV) analyzes static images of signatures to distinguish between genuine and forged samples. To create training data for such systems, negative samples, commonly known as random forgeries, are typically drawn from genuine signatures of other users, as real-world datasets often lack actual forgeries. While this strategy helps address the scarcity of forgery data, it faces several challenges. The randomly selected samples may lack the diversity and challenge needed to improve model robustness. Additionally, they can cause redundancy, increasing training time and storage requirements, and may introduce bias across users, leading to unfair training distributions. This paper proposes a novel strategy, called *ProtoSig*, for generating more informative and diverse negative samples by leveraging prototypical signatures, which are compact, non-identifiable vectors obtained through a data-driven summarization of signature feature vectors. Our experiments demonstrate that ProtoSig enhances skilled forgery detection in a writer-dependent verification approach, eliminating performance variability across runs, while reducing dependence on external user data. We further demonstrate that our method achieves comparable or higher accuracy with a smaller training set, yielding substantial scalability benefits: over 98% reduction in training time and up to two orders of magnitude lower computational cost (in FLOPs), while strengthening data privacy and promoting fairness in offline HSV systems.

1. Introduction

Handwritten Signature Verification (HSV) systems aim to differentiate between genuine signatures, belonging to an individual, and forgeries crafted by impostors. These systems are divided into two main categories: online and offline [1]. Online HSV relies on dynamic features like pressure and speed captured during the signing process, whereas offline HSV analyzes static signature images, which presents a more complex challenge due to the lack of temporal information [2].

Two approaches can be employed in offline HSV systems: writer-dependent (WD) and writer-independent (WI). In the WD approach, a unique classifier is trained for each enrolled user, likely delivering higher performance but increasing complexity due to the need for individual training data for each new user. Conversely, WI systems employ a single, generalized classifier for all users. While this reduces system maintenance and model complexity by facilitating generalization to previously unseen signatures and writers, it often leads to diminished performance [1].

A significant challenge in HSV arises from the limited availability of forgeries during training [2]. Systems often rely solely on genuine signatures to create training datasets and are expected to detect forgeries without directly learning their characteristics. This task is particularly challenging for skilled forgeries, which are carefully created by impostors who know the user's signature [1]. A common approach is to use genuine signatures from other users, known as random forgeries, to represent the negative class [3–12]. This process is illustrated in Fig. 1(top). After being acquired, signature images are converted into feature vectors using a pre-trained representation model. The resulting vectors are stored in a database and later employed for generating training data. To this end, some signature feature vectors of each user are selected to form the positive class, while some feature vectors from other users are randomly selected to form the negative class.

This approach requires that third-party signatures be accessible whenever a new model is created, so that a diverse set of samples can comprise the negative class for classifier training. While increasing the number of users and samples enhances diversity, which is not always possible due to data scarcity, it also introduces a dependency on dataset size, directly impacting system scalability and resource requirements. Moreover, reliance on external signatures raises privacy concerns that can affect development, particularly in security-sensitive

* Corresponding author.
E-mail addresses: kecia.gomes-de-moura.1@ens.etsmtl.ca (K.G. de Moura), robert.sabourin@etsmtl.ca (R. Sabourin), rafael.menelau-cruz@etsmtl.ca (R.M.O. Cruz).

https://doi.org/10.1016/j.patcog.2026.114258
Received 3 October 2025; Received in revised form 7 June 2026; Accepted 9 June 2026
Available online 15 June 2026
0031-3203/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

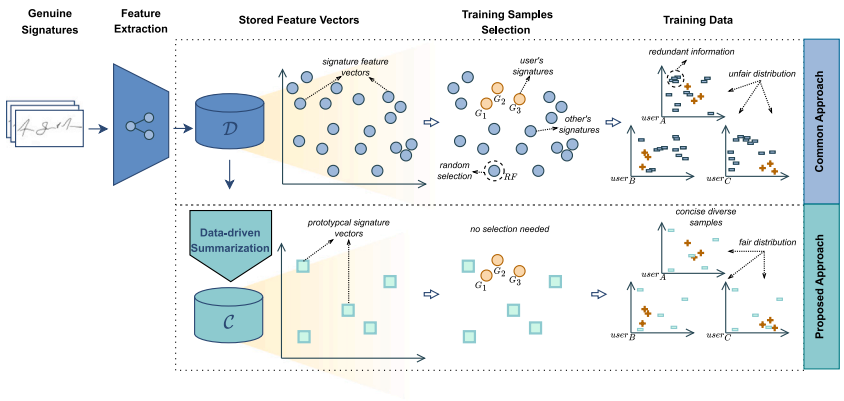


Fig. 1. Comparison between the common HSV approach (top) and the proposed ProtoSig method (bottom) for training data generation. In the common approach, features extracted from signature images are stored in a development set (D), from which random forgeries (RF) are arbitrarily selected with no relation to genuine signatures (G) of the target user. This often produces redundant, unevenly distributed negative samples. In contrast, our method applies a data-driven summarization to D, producing a compact set of representative vectors stored in the summary set (C). These *prototypical signatures* capture the overall distribution of signatures, yielding fewer but more diverse negative samples.

contexts where storing or sharing biometric data is restricted. Furthermore, the stochastic nature of the standard sampling approach may yield inconsistent sample quality across runs, leading to performance variability and an unfair distribution of training samples across users. An effect that, to the best of our knowledge, has not been previously studied.

The limitations of random selection underscore the need to identify more informative training samples in order to improve system performance while ensuring consistent results. Nonetheless, identifying such samples, especially those that closely resemble a given user's signature, can be time-consuming or even impractical, as it would require comparisons across all other users' signatures in the dataset. Given these challenges, we formulate the following general research question: *How can we generate more informative training samples to improve and ensure consistent classifier performance, while promoting fairness across users and supporting scalable, privacy-preserving model development*? To answer this question, we propose a novel data-driven summarization approach for training data generation called *ProtoSig*. The proposed method reduces data storage and retrieval overload while minimizing the reliance on external handwritten signatures. The concept of the proposed approach is illustrated in Fig. 1(bottom).

ProtoSig summarizes the entire set of signature feature vectors by partitioning them into distinct groups that aggregate samples with similar characteristics. This process produces a reduced set of representative vectors that capture the underlying distribution of the development data. These vectors are called *prototypical signatures* and serve as higher-quality, more diverse negative samples, ultimately improving the classifier's ability to detect skilled forgeries. Additionally, the summarization process is executed only once on the development set. It produces non-identifiable feature vectors rather than real signatures, making them reusable across different users and scenarios while also preserving data privacy.

To validate the proposed method, we conducted experiments in a writer-dependent setting using three datasets: GPDS Synthetic, CEDAR, and MCYT-75. The generated training data was used to train a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel. Results show that the proposed method can be seamlessly integrated into existing HSV systems, improving or maintaining accuracy while significantly reducing storage requirements and computational costs, thus providing a practical solution for real-world deployments.

The main contributions of this work are as follows:

- A novel method based on prototypical signatures for generating informative samples for HSV training.
- A training sampling strategy that significantly reduces variability in model performance and ensures fairness in sample selection across users.
- A training-size reduction solution for HSV systems that mitigates computational cost and complexity.
- We show that our approach strengthens data privacy and security by replacing raw signatures with compact, non-identifiable prototypical samples, eliminating the need to store or access signatures belonging to other individuals.
- We demonstrate that our method enables multi-dataset training, allowing prototypical signatures to be learned from combined datasets and thereby capturing generalizable patterns across different signature distributions.

2. Related works

Recent approaches in offline handwritten signature verification differ in their primary objectives, focusing on different areas such as: representation learning [4–6,8,11–16], where the goal is to learn more discriminative features or improve backbone architectures for signature embeddings; data synthesis and/or augmentation [3,9,10], in which new samples, either artificial or distorted, are generated to increase data availability; self-supervised learning (SSL) [17,18], which aims to decrease dependency on labeled data; stream-based or incremental learning [19], among others. Table 1 summarizes the key characteristics of these works.

Despite their differences, most systems share a common approach to obtaining negative samples, which we classify into three categories: (1) traditional random forgeries (RF), genuine signatures from other users; (2) skilled forgeries (SK), human-produced forged signatures; and (3) synthetic forgeries (SY), algorithmically generated samples.

Most works rely on standard RF selection. Tsourounis et al. [4] propose transfer learning using metric learning to adapt a CNN for signature representation in a WD setup with SVM-RBF. Tsourounis et al. [6] extend this via feature-based knowledge distillation (FKD), and Tsourounis et al. [16] further extend the FKD framework with graph-based geometric regularization, achieving state-of-the-art performance in both WI and WD settings. Viana et al. [14] introduce a continual learning framework that integrates synthetic and real signature data via knowledge distillation, training a WD SVM-RBF with random forgeries. Other representative works include Zhang et al. [8], who propose a VAE with disentanglement loss for feature extraction; Li et al. [7], a transformer-based model evaluated in both WD and WI settings; and Zheng et al. [12], a hybrid transformer-CNN network, all using traditional RF sampling.

A smaller number of works incorporate skilled forgeries during training. Xiong et al. [15] introduce an attention-based multi-siamese network (MSN) trained in a WI setup, while Ren et al. [13] propose a

Table 1
Summary of recent offline HSV works, detailing primary focus, structure, verification setting (WI/WD), representation model, classifier, and negative sampling strategy: using random forgeries (RF), skilled forgeries (SK), or synthetic forgeries (SY).

Work	Focus	Structure	WD/WI	Extractor	Classifier	Kernel	Negative samples
Prajapati et al. 2021 [3]	Synthesis	End-to-end	WD	CNN	ANN	n/a	Traditional RF + SY
Chattopadhyay et al. 2022 [17]	Representation	Modular	WI	ResNet-18	Distance Metric	n/a	Uses SK + Traditional RF
Manna et al. 2022 [18]	Representation	Modular	WI	ResNet-18	SVM	RBF	Uses SK + Traditional RF
Tsourounis et al. 2022 [4]	Representation	Modular	WD	CNN	SVM	RBF	Traditional RF
Longjam et al. 2023 [5]	Representation	Modular	WI	CNN-BiLSTM	BiLSTM	n/a	Traditional RF
Ren et al. 2023 [13]	Representation	End-to-end	WI	2C2S	Fusion Decision (FD)	n/a	Uses SK
Tsourounis et al. 2023 [6]	Representation	Modular	WD	SigNet, ResNet-18	SVM	RBF	Traditional RF
Li et al. 2024 [7]	Representation	Both	Both	TransOSV	SVM	Linear and RBF	Traditional RF
Viana et al. 2024 [14]	Representation	Modular	WD	SigNet-CL	SVM	RBF	Traditional RF
Xiong et al. 2024 [15]	Representation	End-to-end	WI	MSN	Voting	n/a	Uses SK
Zhang et al. 2024 [8]	Representation	Modular	WD	VAE	SVM	RBF	Traditional RF
Arab et al. 2025 [9]	Synthesis	Modular	WD	SigNet, LDP, HOT	SVM	RBF	Traditional RF
Arab et al. 2025 [10]	Synthesis	Modular	WD	SigNet, LDSP	SVM	RBF	Traditional RF
Tsourounis et al. 2025 [16]	Representation	Modular	Both	SigNet, ResNet-18	SVM, DSC-BFS	RBF	Traditional RF
Zhao et al. 2025 [11]	Representation	Modular	WD	DCCCAE	SVM	RBF	Traditional RF
Zheng et al. 2025 [12]	Representation	Modular	Both	HTCSigNet	SVM	RBF	Traditional RF
Zois et al. 2025 [20]	Synthesis	Modular	WI	SPD matrix	Metric learning	n/a	Uses SY

Abbreviations: Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (BiLSTM), Variational Autoencoder (VAE), Multi-Siamese Network (MSN), Two-Channel Two-Stream (2C2S), Deep Canonically Correlated Contractive Autoencoder (DCCCAE), Local Directional Patterns (LDP), Histogram of Templates (HOT), Gentle AdaBoost Feature Selection (BFS), Decision Stump Committee (DSC) DSC-BFS. Symmetric Positive Definite (SPD).

two-channel, two-stream (2C2S) transformer-based network leveraging skilled forgeries to enhance WI verification. Other works [17,18] adopt self-supervised learning (SSL) for creating representation models and performing classification using traditional RF and skilled forgeries

Other methods focus on generating synthetic forgeries. Prajapati et al. [3] use an autoencoder to introduce distortions in genuine signatures, Arab et al. [9] and Arab et al. [10] adopt handcrafted and deep features to generate synthetic positive signatures, training radial basis function SVMs under WD settings. Likewise, Zhao et al. [11] fuse deep learning-based and hand-crafted features to train an SVM-RBF with RF as negatives. More recently, Zois et al. [20] propose a WI framework using Riemannian Gaussian mixtures to model signatures and generate synthetic positive and negative samples. Although the authors also employ unsupervised learning in their method, they rely on distributional assumptions and explicit sample generation, whereas our approach focuses on condensing real population data into a compact set of representative samples via clustering.

Despite their differences in architecture and learning strategy, the majority of recent HSV systems continue to rely on the traditional random forgery strategy for constructing the negative class, either as a stand-in for real forgeries or in addition to synthetic ones. Most of them are focused on *representation learning*, aiming to design better feature extractors or embeddings for signatures (denoted as "representation" in Table 1). In contrast, we focus on improving the generation of training data for signature verification while leveraging existing representation models. This model-agnostic method can be readily adopted by a wide range of HSV systems that use RF sampling to achieve improved results.

3. Proposed method

An essential part of an HSV system is generating training data, which has a direct impact on classifier performance. In this work, we introduce *ProtoSig*, a novel data generation strategy designed to enhance the selection of negative samples. The overview of the method is presented in Fig. 2. A list of symbols and notation is provided in Section 2 of the supplementary material.

The process begins with a collection of M offline handwritten signature images $S = \{S_1, S_2, \dots, S_M\}$ from N distinct users. A feature extraction model $\phi(\cdot)$ maps each $S_i \in S$ to a feature vector, producing $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ with $\mathbf{x}_i \in \mathbb{R}^d$, collected into a development set D. Subsequently, D undergoes a data-driven summarization $K(\mathcal{X}, k)$, which partitions the feature space into k regions and computes prototypical signatures $C = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k\}$ that reflect the overall data distribution. The prototypical signatures, along with samples from D, are then retrieved and combined to construct the desired training set $\mathcal{T}$.

3.1. Data-driven summarization

The underlying intuition of the proposed approach is illustrated in Fig. 3. In this example, we consider $N = 10$ users, each contributing $m = 5$ handwritten signatures, resulting into a collection of $M = N \times m = 50$ samples represented in a two-dimensional feature space. One way to improve the standard approach of arbitrary random forgery selections as negative samples is to select the most informative samples. Considering that skilled forgeries are recognized as the most challenging type of forgeries to detect due to their high visual similarity to genuine signatures [1], we hypothesize that selecting random forgeries that are closest to a given genuine sample A (from the user for whom we intend to build a model), as illustrated in Fig. 3(a), can improve the classifier's ability to learn more discriminative decision boundaries. In the context of HSV, this approach is regarded as a hard negative mining process [21].

However, exhaustively searching for the closest samples across all other users' signatures (Fig. 3(b)) becomes computationally infeasible, particularly in the WD setting where the process must be repeated for each enrolled user. Moreover, focusing exclusively on hard samples may lead to overfitting [22] and can overlook the representative diversity needed for fair and robust training [23,24].

The summarization process (Fig. 3(c)) addresses these limitations by identifying prototypical signature vectors that compactly represent the feature space, eliminating the need to compute per-user distances (Fig. 3(d)). Since signatures from different users may be grouped together, the prototypical vectors capture a diverse set of characteristics, balancing informativeness and diversity. Furthermore, replacing random sampling with our method is expected to make the sample selection process more deterministic compared to the inherent stochasticity of random sampling. *We hypothesize that our technique leads to greater consistency and control in training data generation, potentially reducing variability across runs*. In other words, our approach may help ensure that all users receive training sets with more comparable representativeness, which could reduce the risk of bias introduced by random fluctuations in sample quality. This consideration is particularly important in scenarios with limited signature data, where random sampling may lead to high variance in model performance across different runs.

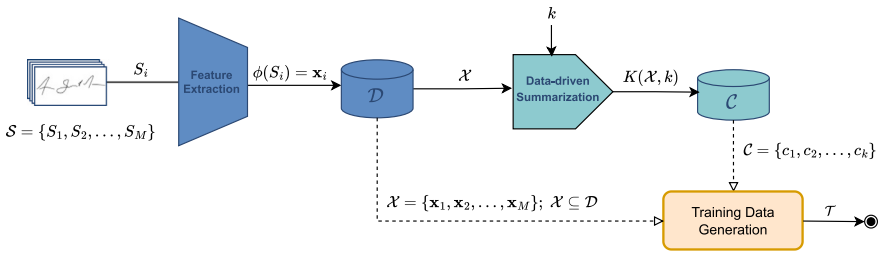
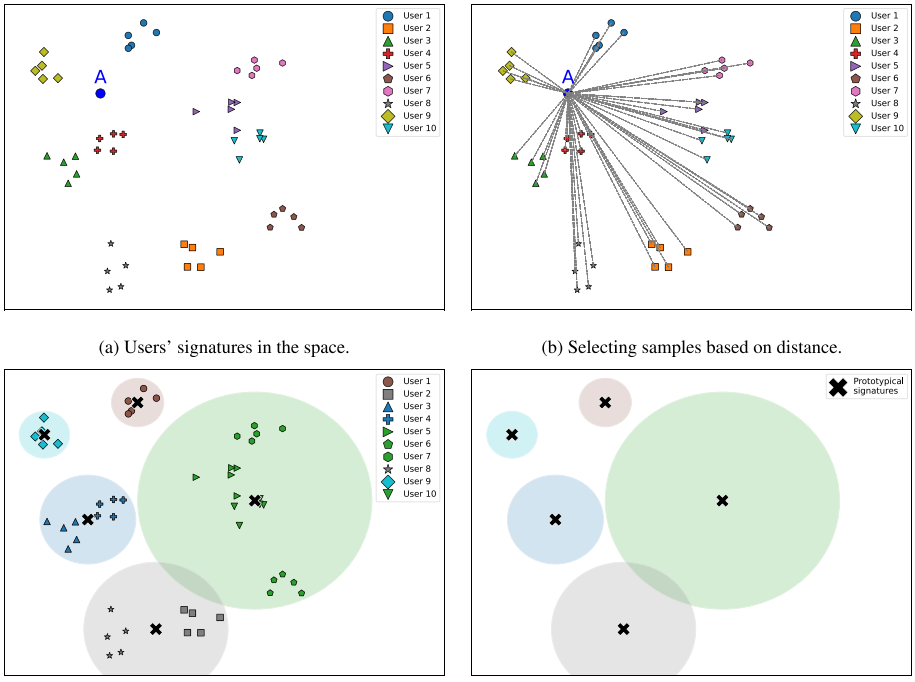


Fig. 2. Overview of the proposed method. A set of handwritten signature images $\mathcal{S} = \{S_1, \ldots, S_M\}$ is transformed into feature vectors $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}$ via a feature extractor $\phi(\cdot)$. These vectors form the development set D, which undergoes data-driven summarization $K(\mathcal{X}, k)$ to produce prototypical signatures $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$. The prototypes are then used to generate training data $\mathcal{T}$ for classifier learning.



(a) Users' signatures in the space.

(b) Selecting samples based on distance.

(c) Example of summarization.

(d) No selection needed with summarization.

Fig. 3. Visual comparison of negative samples selection based on distance and based on data-driven summarization. (a) Signature feature vectors from multiple users. (b) Distance computation for more informative samples; likely impractical in real-world scenarios. (c) Example of summarization, reducing the feature space to a representative subset. (d) The reduced set is used as negative samples, eliminating the need for computing distances.

The summarization employs a clustering algorithm on the full feature space. The clustering groups underlying patterns of genuine signatures from a population of users. In this work, we investigate multiple clustering algorithms in order to broaden the summarization process across different families of clustering methods. The resulting set of prototypical samples, denoted as $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$, provides a compact and representative abstraction of the original data distribution in the development set.

3.2. Training data generation

The training data generation process is illustrated in Fig. 4, with the detailed steps in Algorithm 1 of the supplementary material. The procedure consists of two main stages: (1) retrieving the genuine signature feature vectors for each user from an exploitation set ($\mathcal{E}$), and (2) combining them with the prototypical signatures.

Formally, let g denote the number of signatures available per user in $\mathcal{E}$ for training, and define the set of genuine feature vectors of user j as $\mathcal{X}_j = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_g\}_j$ with $j = 1, \ldots, N$. The prototypical signatures obtained from the summarization process are represented by $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$. These serve as negative samples, while the genuine signatures in $\mathcal{X}_j$ serve as positive samples. The training set for user j is thus constructed as shown in Eq. (1).

$$\mathcal{T}_j = \{(\mathbf{x}, +) \mid \mathbf{x} \in \mathcal{X}_j\} \cup \{(\mathbf{c}, -) \mid \mathbf{c} \in C\}, \tag{1}$$

where $+$ and $-$ denote the class labels for genuine and prototypical signatures, respectively. This dataset is then used to train the writer-dependent classifiers.

A significant advantage of the proposed system is that, once the set of prototypical signature vectors C is generated, the original signatures of external users are no longer needed. This enhances data privacy by avoiding the storage and reuse of raw signature data from external users.

4. Experimental setup

Datasets and segmentation. Experiments were conducted using the datasets detailed in Table 2, each segmented into a development set D, for classifier training, and an exploitation set $\mathcal{E}$, for testing.

Preprocessing. We follow the steps described in [21]: signature images are centered on a canvas sized to the dataset's largest sample; Otsu's

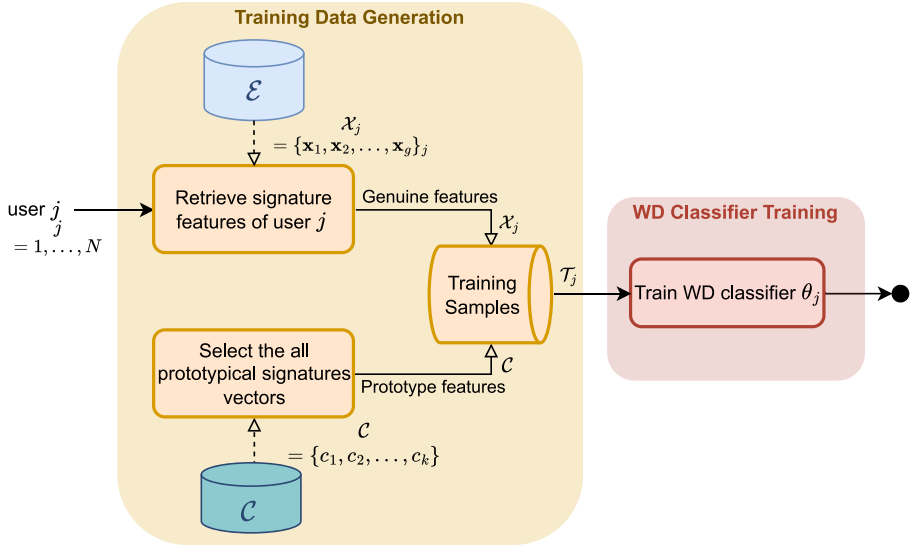


Fig. 4. Overview of the training data generation (yellow) using prototypical signatures for classifier training (red). For each user j, the corresponding set of genuine signature features $\mathcal{X}_j = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_g\}_j$ are retrieved from an exploitation set $\mathcal{E}$. The set of all prototypical signatures $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$ is also retrieved and combined with $\mathcal{X}_j$ to construct the training set $\mathcal{T}_j$, which is used to train a writer-dependent classifier θ_j.

Table 2
Summary of datasets and their user segmentation. G: genuine signatures. SK: skilled forgery signatures.

Dataset	Signatures		Development set $\mathcal{D}$		Exploitation set $\mathcal{E}$	
	Users	G/SK	Users	ID range	Users	ID range
CEDAR [25]	55	24/24	27	1–27	28	28–55
MCYT-75 [26]	75	15/15	37	1–37	38	38–75
GPDS Synthetic [27]	10 000	24/30	581	301–881	300	1–300

algorithm removes the background; the image is inverted, resized to 170 × 242 pixels, and center-cropped to 150 × 220 pixels.

Feature Extraction. We employ SigNet Synthetic (*SigNet-S*) [21], a deep convolutional neural network producing a 2048-dimensional feature vector. *SigNet-S* was originally trained on the GPDS Synthetic dataset [27] using a disjoint subset of users (IDs 5001–7000), ensuring independence from the data employed in our experiments.

Classifier. We employ an SVM-RBF classifier using the `scikit-learn` library [28], with regularization parameter $C = 1.0$ and kernel coefficient $\gamma = 2^{-11}$, following prior work [14,16,21].

Data generation. The writer-dependent approach requires that both training and testing contain samples from the same set of users. Accordingly, the exploitation set $\mathcal{E}$ is divided into training ($\mathcal{E}_L$) and test ($\mathcal{E}_T$) subsets, as detailed in Table 3. To generate positive training samples, between 1 and 12 genuine signatures are randomly selected per user from $\mathcal{E}$. For negative samples, three different strategies are employed: our ProtoSig method and two baseline approaches as described below.

We compare our method against two baseline strategies: the leave-one-writer-out (LOO) approach, which is commonly used in the literature [8,11,12,16,21,29]; and the multi-dataset (multi) strategy, which is included to ensure a fair comparison, as it employs the same training set as our method. In LOO, negative signatures are collected from all other users, reaching several thousand in large datasets, thereby increasing storage requirements and computational complexity. In the multi-dataset configuration, all development sets are merged and negative samples are randomly selected from the combined set. In ProtoSig, summarization is applied to the combined development set, yielding k prototypical samples that serve as negatives during training. Finally, all methods are evaluated independently on each dataset using the corresponding test set $\mathcal{E}_T$.

Clustering algorithms for summarization. We evaluate ProtoSig using clustering algorithms from distinct methodological families, including partition-based, distribution-based, hierarchical, density-based, and representation-learning approaches. Specifically, we considered *K-means*, *Gaussian Mixture Models (GMM)*, *Agglomerative Hierarchical Clustering*, *Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)*, *Density-Based Spatial Clustering of Applications with Noise (DBSCAN)*, and *Unsupervised Dictionary Learning (DL)*.

Hyperparameter selection. For each clustering technique, hyperparameters were selected through cross-validation on the development set. Optimal configurations are summarized in Table 4. The complete validation procedure is detailed in Section 4 of the supplementary material. A sensitivity analysis on k is provided in Section 5, showing that performance remains stable across a wide range of values.

Performance evaluation. System performance is assessed using the Equal Error Rate (EER), defined as the point where the False Rejection Rate (FRR) equals the False Acceptance Rate (FAR) [30]. All experiments respect the fixed $\mathcal{D}$ and $\mathcal{E}$ segmentation and are repeated five times, resampling genuine signatures in each run. EER is computed with both global and user-specific thresholds, and reported as mean ± standard deviation.

Model complexity evaluation. Computational complexity is assessed via training and testing time (seconds), model size (MB),[1] number of support vectors, and training/testing FLOPs (floating-point operations), computed using the PAPI[2] library [31] via PYPAPI.[3] Machine configuration details are provided in Section 6 of the supplementary material.

5. Results

This section presents experimental evidence supporting our main claims: (1) ProtoSig matches or outperforms standard random sampling; (2) it reduces training size and computational requirements; (3) it enhances training stability by eliminating performance variability from stochastic sampling; (4) it improves fairness across users; and (5) it enhances data privacy.

[1] In WD settings, where a classifier is trained for each user, model size refers to the total size obtained by summing the sizes of all individual models.
[2] https://github.com/icl-utk-edu/papi
[3] https://github.com/flozz/pypapi

Table 3
Data segmentation and negative sample configuration. For each dataset, g genuine signatures are selected for training, and the negative samples follow different strategies depending on the method.

Dataset	Training signatures			Testing signatures ($\mathcal{E}_T$)
	Genuine ($\mathcal{E}_L$)	Negative		
		Samples	Method	
CEDAR	$g \in \{1, 2, 3, 5, 10, 12\}$	$g \times 54$ ($\mathcal{D} \cup \mathcal{E}$) $g \times 645$ ($\mathcal{D}$) k prototypical signatures ($\mathcal{D}$)	Standard (LOO) Standard (multi) ProtoSig	10 genuine, 10 skilled
MCYT	$g \in \{1, 2, 3, 5, 10\}$	$g \times 74$ ($\mathcal{D} \cup \mathcal{E}$) $g \times 645$ ($\mathcal{D}$) k prototypical signatures ($\mathcal{D}$)	Standard (LOO) Standard (multi) ProtoSig	5 genuine, 5 skilled
GPDS-S	$g \in \{1, 2, 3, 5, 10, 12\}$	$g \times 299$ ($\mathcal{E}$) $g \times 645$ ($\mathcal{D}$) k prototypical signatures ($\mathcal{D}$)	Standard (LOO) Standard (multi) ProtoSig	10 genuine, 10 skilled

Table 4
Cross-validation results for each clustering algorithm.

Method	Type	Parameter(s)	Optimal
K-means	Partitional	Number of centroids	$k = 50$
GMM	Distributional	Gaussian components	$k = 50$
Agglomerative	Hierarchical	Ward's linkage clusters	$k = 50$
BIRCH	Hierarchical	Cluster radius threshold	$t = 15$
DBSCAN	Density-based	Neighbor radius, min samples	$\epsilon = 40$, $min_s = 5$
Dictionary Learning	Representation	Sparsity regularization, atoms	$\alpha_r = 0.5$, a = 10

Table 5
Skilled forgery detection using ProtoSig with different clustering methods across a varying number of genuine training signatures (g). Results show EER (%) with a global threshold. Prototypical signatures are obtained via multi-dataset training combining CEDAR, MCYT, and GPDS-S.

Data	g	K-means	GMM	Agglomerative	BIRCH	DBSCAN	DL
CEDAR	1	19.50 ± 2.13	19.46 ± 2.09	19.50 ± 2.02	**19.14 ± 1.99**	19.75 ± 1.75	50.00 ± 0.00
	2	13.57 ± 1.63	13.54 ± 1.65	13.46 ± 1.76	**13.07 ± 0.88**	13.71 ± 1.44	49.96 ± 0.07
	3	11.57 ± 1.22	11.43 ± 1.09	11.64 ± 1.46	**11.18 ± 1.33**	12.04 ± 1.21	48.64 ± 1.17
	5	9.14 ± 0.36	9.21 ± 0.38	9.18 ± 0.31	**9.07 ± 0.21**	9.93 ± 0.24	47.43 ± 1.44
	10	6.75 ± 0.29	**6.71 ± 0.29**	6.89 ± 0.43	**6.71 ± 0.40**	7.54 ± 0.56	40.54 ± 1.27
	12	**6.11 ± 0.97**	6.18 ± 0.93	6.18 ± 0.79	6.50 ± 0.86	7.39 ± 1.05	35.61 ± 1.00
MCYT	1	17.37 ± 1.62	17.11 ± 1.60	17.00 ± 1.63	**17.00 ± 2.11**	18.26 ± 1.93	50.00 ± 0.00
	2	11.95 ± 1.37	12.05 ± 1.55	11.95 ± 1.95	**11.42 ± 1.95**	12.26 ± 1.48	50.00 ± 0.00
	3	10.26 ± 1.03	**10.05 ± 1.16**	10.42 ± 1.26	10.32 ± 1.35	10.47 ± 1.39	50.00 ± 0.00
	5	**6.37 ± 1.19**	**6.37 ± 1.19**	6.47 ± 1.53	7.47 ± 1.11	7.16 ± 1.17	50.00 ± 0.00
	10	6.21 ± 0.49	**6.11 ± 0.56**	6.21 ± 0.59	**6.11 ± 0.61**	6.74 ± 0.27	46.00 ± 1.08
GPDS-S	1	16.59 ± 0.46	16.64 ± 0.44	16.63 ± 0.46	17.07 ± 0.48	**16.06 ± 0.50**	50.00 ± 0.00
	2	12.02 ± 0.33	12.03 ± 0.36	12.09 ± 0.29	12.10 ± 0.50	**11.58 ± 0.30**	50.00 ± 0.00
	3	9.96 ± 0.16	9.99 ± 0.13	10.02 ± 0.15	10.09 ± 0.22	**9.78 ± 0.17**	49.99 ± 0.01
	5	7.81 ± 0.23	7.85 ± 0.24	7.86 ± 0.21	8.13 ± 0.26	**7.77 ± 0.21**	49.77 ± 0.07
	10	6.13 ± 0.15	6.11 ± 0.11	**6.11 ± 0.09**	6.41 ± 0.10	6.29 ± 0.16	47.99 ± 0.14
	12	**5.63 ± 0.21**	5.64 ± 0.22	5.66 ± 0.23	5.96 ± 0.18	5.68 ± 0.19	46.55 ± 0.13

5.1. Impact of clustering choice

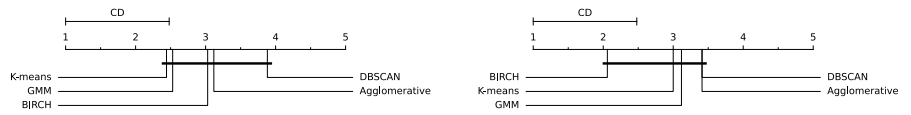
For each clustering method, prototypical vectors were derived using the hyperparameters that yielded the best performance in the validation process (see Table 4): cluster centroids for K-means ($k = 50$), Gaussian components for GMM ($k = 50$), mean vectors for Agglomerative ($k = 50$), BIRCH ($t = 50$), and DBSCAN ($\epsilon = 40$), and dictionary atoms for DL (a = 10). These vectors were used as negative samples to train WD SVM-RBF classifiers, with results reported in Table 5 using global thresholds. Results for user-specific thresholds are provided in Section 7 of the supplementary material.

Most clustering strategies yield comparable verification performance. The only exception is Dictionary Learning, which consistently operates near chance level, as it is primarily designed for sparse signal reconstruction rather than data partitioning [32].

A Friedman test ($\alpha = 0.05$) was conducted, excluding the DL algorithm, to assess statistically significant differences in clustering results. The Friedman test ranks each clustering method according to its EER across all evaluated subjects and training configurations, considering both global and user-specific thresholds. Experiments were conducted on the GPDS-S, CEDAR, and MCYT datasets using different numbers of genuine samples (g) for training. In total, 17 distinct training scenarios were evaluated (3 datasets × 6 or 5 different g), in which the EER constitutes a repeated measure across clustering methods. In all cases, the null hypothesis of equivalent performance among methods was not rejected at the 5% significance level (p = 0.058 for global thresholds and p = 0.070 for user-specific thresholds).

For completeness, pairwise comparisons were examined using the Nemenyi procedure, and the corresponding Critical Difference (CD) diagrams are presented in Fig. 5, in which the left-most positioned methods correspond to better performance, while methods connected by a horizontal bar are not statistically distinguishable.

Results show that K-means achieves the best average ranking under global thresholds, and that BIRCH achieves the best average ranking under user-specific settings, but no statistically significant pairwise differences were identified. These results indicate that ProtoSig's performance gains stem from the summarization principle itself rather than



(a) Global thresholds. (b) User-specific thresholds.

Fig. 5. Critical Difference (CD) diagrams comparing ProtoSig performance using different clustering algorithms for the summarization step, under (a) global and (b) user-specific thresholds.

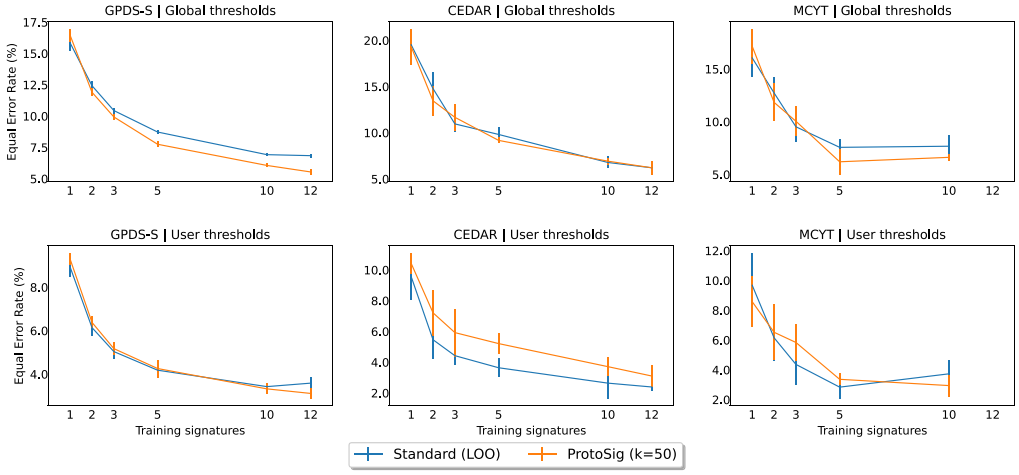


Fig. 6. Average EER for skilled forgery detection across different training signatures using the optimal k for K-means summarization, compared against the LOO standard approach.

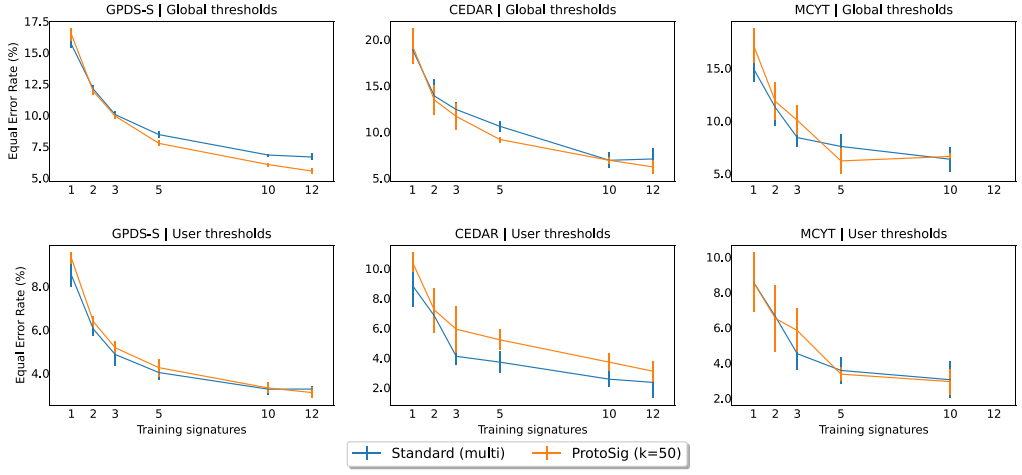


Fig. 7. Average EER for skilled forgery detection across different training signatures using the optimal k for K-means summarization, compared against the multi-dataset standard approach.

from a particular clustering technique. We therefore adopt *K-means* throughout the subsequent evaluation.

5.2. ProtoSig vs. Standard sampling performance

Figs. 6 and 7 show results for the LOO and multi-dataset configurations, respectively. Our method for generating negative samples matches or surpasses the performance of both baselines when using an SVM-RBF classifier. Tabular results including FAR and FRR metrics are provided in Section 7 of the supplementary material.

These results are achieved while significantly reducing training size, as shown in Fig. 8. At maximum training signatures, the LOO configuration requires 3588 negative samples for GPDS-S, 648 for CEDAR, and 740 for MCYT, while the multi-dataset approach requires up to 7740. In contrast, ProtoSig uses only 50 negative samples regardless of configuration, demonstrating a substantial reduction in training data requirements while also addressing the limited-data problem in smaller datasets via multi-dataset training.

5.3. ProtoSig minimizes computational cost

Fig. 9 compares computational metrics of SVM-RBF trained with standard random forgeries versus prototypical signatures ($k = 50$). ProtoSig significantly alleviates the computational burden, evidenced by a near-constant cost as the user count increases, compared with the linearly increasing cost of the standard method.

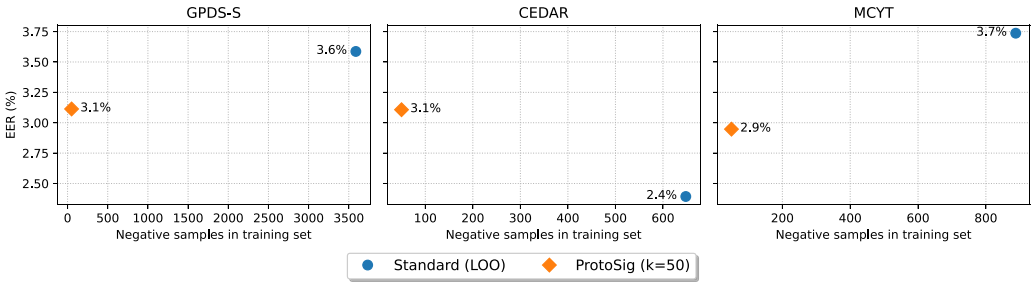


Fig. 8. Average EER vs. the total number of negative samples in the training set for the standard approach (circles) and ProtoSig (diamonds), using the maximum number of genuine signatures and user-specific thresholds.

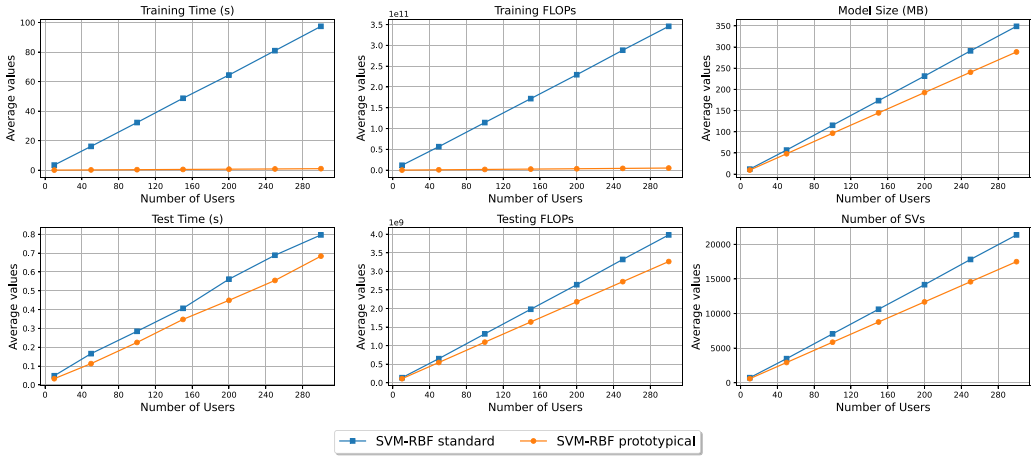


Fig. 9. Computational cost of SVM-RBF with standard versus prototypical training data (K-means, $k = 50$) for WD signature verification on GPDS-S, plotted against an increasing number of users.

Table 6
Computational cost comparison between standard and prototypical ($k = 50$) approaches for the largest GPDS-S configuration.

Metric	Standard	Prototypical	Metric	Standard	Prototypical
Training Time (s)	97.39	1.01	**Testing Time (s)**	0.80	0.68
Model Size (KB)	348,856.21	288,401.10	**Number of SVs**	21,362	17,498
Training FLOPs	3.46×10^{11}	5.08×10^{9}	**Testing FLOPs**	3.98×10^{9}	3.26×10^{9}

Table 6 provides a detailed comparison at maximum user count. As shown, ProtoSig reduces training time by more than 98× and decreases training FLOPs by approximately two orders of magnitude.

5.4. *ProtoSig mitigates performance variability*

We investigated whether the randomness of standard negative-sample acquisition affects verification results at the individual-user level. This perspective is important in practice, as each user in a real-world system typically has their own dedicated model.

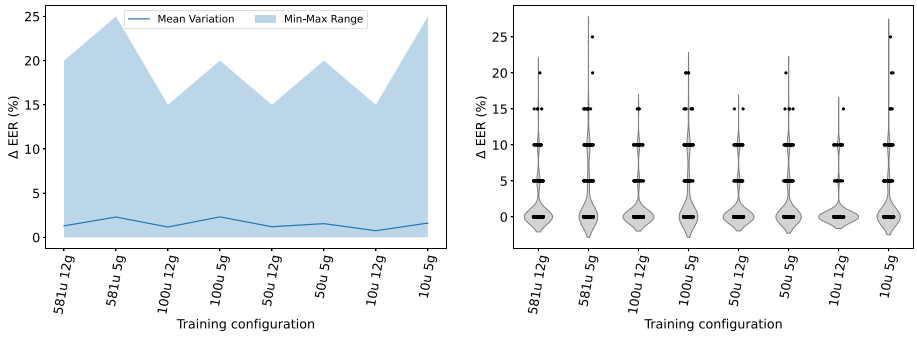
We fixed the set of genuine training samples across runs and obtained negative samples using the standard random sampling approach. Experiments were conducted on GPDS-S under various training configurations, with different numbers of users (u) and genuine signatures (g). Specifically, we subsegmented the GPDS-S development set into subsets of varying sizes: 10 users (IDs 301–310), 50 users (301–350), 100 users (301–400), and the full 581 users (301–881). The exploitation set $\mathcal{E}$ comprised 300 users (IDs 1–300). For each user in $\mathcal{E}$, we selected g genuine signatures, $g \in \{5, 12\}$, and a total of $g \times u$ random forgeries from the chosen D subset to train an SVM with RBF kernel. Tests were then performed on $\mathcal{E}$ using 10 genuine and 10 skilled forgery signatures.

After inference, we computed the EER at the individual level, repeating the training and test processes across five runs, with the only source of randomness being the negative sampling. We then recorded the minimum and maximum EER values observed for each user and calculated the average variation in EER. Results are reported in Fig. 10, with Fig. 10(a) showing the average EER range per user for each training configuration, and Fig. 10(b) providing a distributional view of EER variation. Complete analysis on CEDAR and MCYT datasets is provided in Section 8 of the supplementary material.

As shown in Fig. 10(a), the EER for some users varied by more than 20 percentage points across different runs. Additionally, Fig. 10(b) demonstrates that the standard approach results in a wide distribution of user EERs between runs. This high degree of unpredictability poses a significant challenge for deployment, where stable, trustworthy performance is essential.

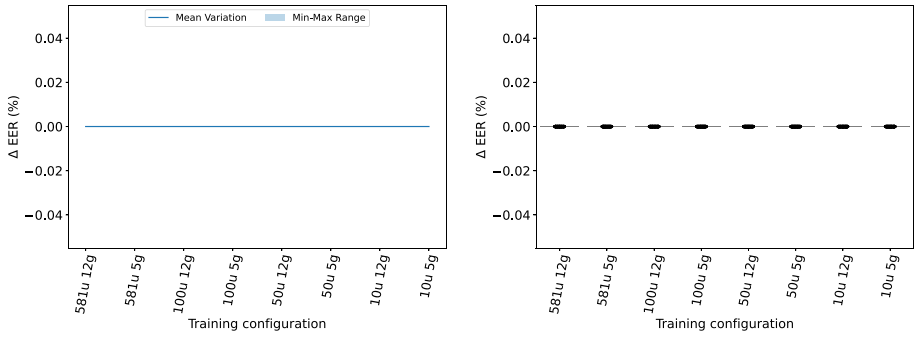
For comparison, we conducted the same set of experiments with ProtoSig, replacing the acquisition of negative samples with the summarization method, employing $k = 50$. Results reported in Fig. 11 demonstrate that our technique eliminates the instability during WD classifier training by using the same set of prototypical negative samples for every training run. While variability may still arise if the summarization procedure is repeated (as different sets of prototypical signatures could be produced), this step is performed only once and reused across users, ensuring stable and consistent training thereafter.

Results in Table 7 better illustrate the impact at the user level, showcasing four users with the widest performance variation under



(a) User EER variation. (b) Distribution of user EER variation.

Fig. 10. EER variation using the standard approach across different training configurations on GPDS-S (u: users, g: genuine signatures per user).



(a) User EER variation. (b) Distribution of user EER variation.

Fig. 11. EER variation using ProtoSig ($k = 50$) across different training configurations on GPDS-S.

Table 7
User EER values across 5 different runs ($g = 12$, $u = 581$).

User ID	Type	EER_1	EER_2	EER_3	EER_4	EER_5
12	Standard	5%	5%	5%	15%	5%
12	ProtoSig	5%	5%	5%	5%	5%
32	Standard	10%	5%	15%	15%	10%
32	ProtoSig	10%	10%	10%	10%	10%
207	Standard	5%	5%	0%	10%	15%
207	ProtoSig	5%	5%	5%	5%	5%
209	Standard	5%	15%	10%	10%	10%
209	ProtoSig	5%	5%	5%	5%	5%

the standard method (when $g = 12$ and $u = 581$). For instance, user 12 exhibited EERs ranging from 5% to 15%, an unpredictable swing that could affect system trust. With the summarization-based approach, however, the EER remained stable at 5%, providing consistent, and reliable performance.

5.5. ProtoSig provides fairer systems

In biometric systems, performance metric values are not the only important factor, but also how this performance is attained among different users. A reliable system should distribute errors evenly across authentic users and impostors, avoiding systematic disadvantages for specific individuals [33].

To assess fairness, we analyzed the distribution of verification errors across users by computing the Lorenz Curves [34] and the Gini Coefficient (GC) [35] metrics. While Lorenz curves and GC were originally introduced to quantify inequality in economic systems, their use in biometric evaluation has been formally motivated in [36] by the need to distinguish between *random* and *systematic* errors across users. In this context, a low Gini coefficient for FAR indicates that false accepts are evenly spread across attackers, reducing the risk of systematic false accepts that would allow specific impostors to repeatedly evade detection. Conversely, a higher Gini coefficient for FRR suggests that false rejections are concentrated among a small subset of users, which is often preferable in practice, as such cases can be investigated, mitigated, or addressed through calibration or alternative authentication strategies [36].

Fig. 12 reports the results for both the standard and proposed method, considering the false acceptance and rejection rates obtained from the experiments described in the previous section on GPDS-S dataset. Complete analysis on CEDAR and MCYT datasets is provided in Section 8 of the supplementary material.

Considering the Gini Coefficient, which corresponds to the area between the Lorenz Curve and the Line of Equality, we observe that the standard approach's GC of 0.81 indicates possible systematic errors in false accepts. At the same time, the proposed method essentially presents random errors given the low GC of 0.48. Having a lower GC for FAR is preferable, as it suggests that false accepts are distributed more uniformly among attackers rather than allowing a small number of individuals to consistently bypass detection [36].

Additionally, the results show that false rejections are more evenly distributed in the standard method (GC = 0.72), whereas they are concentrated among a few users in the proposed method (GC = 0.98). A higher Gini coefficient for FRR suggests that only a small subset of users exhibits erratic or inconsistent behavior, potentially due to factors such as poor system calibration or users not signing properly. In this context, a higher GC is actually preferable, as it indicates that errors are not systemic but rather limited to a few outliers, making the issue easier to isolate and address, either through system design improvements or operational measures [36].

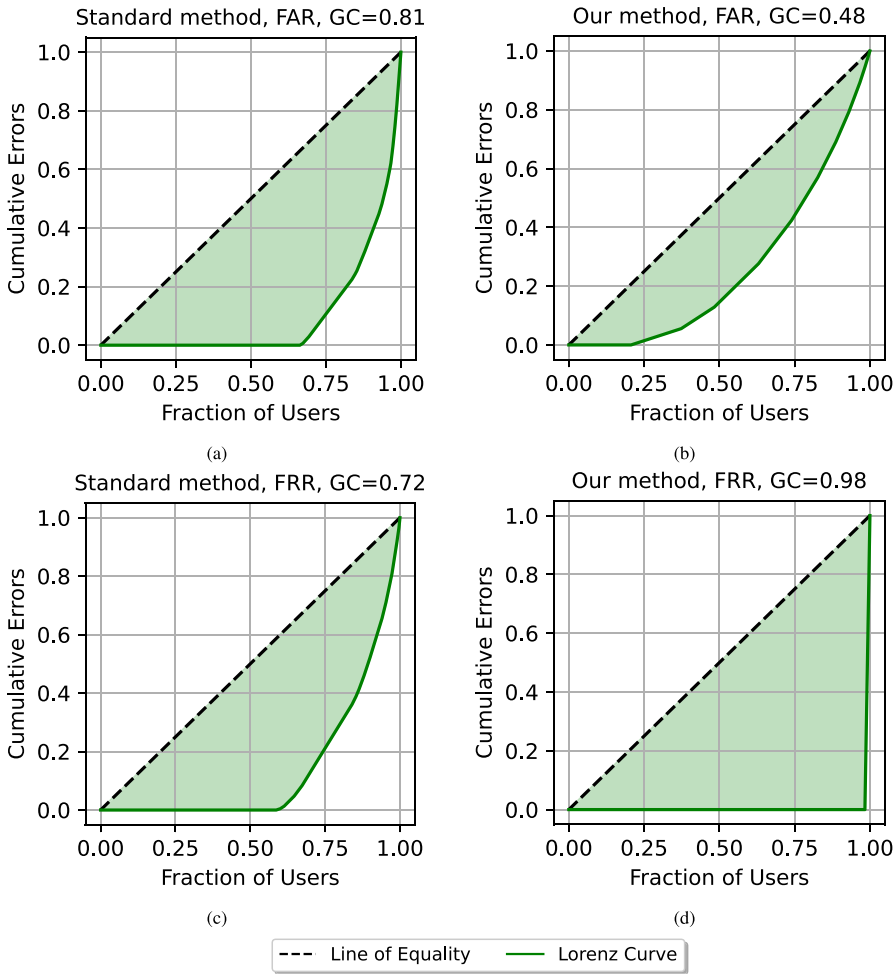


Fig. 12. For false accepts, (a) the standard approach's GC of 0.81 points to systematic errors, (b) while ProtoSig's GC of 0.48 indicates essentially random errors. (c) False rejects are more evenly distributed in the standard method (GC = 0.72), (d) whereas they are concentrated in a small fraction of users for ProtoSig (GC = 0.98), suggesting issues affecting only a few users.

5.6. Comparative evaluation with existing works

An important strength of the ProtoSig training strategy is its potential to be integrated into existing methods, thereby improving the system without requiring substantial modifications to the original architectures. To demonstrate this, we evaluated the effect of applying our method to previously published works. A fair comparison in this context requires analyzing the systems in their original form and comparing them with the adapted configuration that incorporates our technique.

It is important to note that this analysis could only be conducted on works for which code was publicly available and reproducible. Specifically, we focused on two recent studies, [16] and [14], whose implementations are available on GitHub,[4] and for which our reproduced results were consistent with the metrics reported in their original publications. This ensures that any observed differences can be attributed to the integration of our technique rather than inconsistencies in reproduction.

Backbones. In [16], feature knowledge distillation (FKD) was employed to transfer knowledge from a teacher model (SigNet) to a student (ResNet18) without requiring signature images during distillation. The authors explored several KD strategies, such as geometric distillation (GEOM) for local feature alignment, and three global distillation objectives were considered: (i) temperature-scaled cross-entropy (T-CE), (ii) Barlow Twins (BT), encouraging redundancy reduction, and (iii) the proposed Barlow Colleagues (BC), which is a relaxed version of BT that promotes high-correlation alignment without enforcing strict element-wise correspondence. Since the study concluded that combining local (GEOM) and global (classification-oriented) KD is the most effective approach, we adopted the following three backbones: GEOM & TCE, GEOM & BT, and GEOM & BC.

In [14], a continual learning based on knowledge distillation is employed to build representation models more suitable for real signatures verification. The method involves generating inverted examples with the same distribution as real examples and complementing a feature space based on real data using synthetic data. In the process, knowledge is distilled from a teacher (SigNet) into different student models through a loss function that combines Kullback–Leibler divergence and cross-entropy terms. The resulting continual models cover different backbone architectures, all employed in our study: Continual SigNet (AlexNet-based), Continual ResNet152, and Continual Vision Transformer (ViT).

Classification. As both works employ the same configuration for training and testing an SVM with a radial basis function kernel, we used their representation models for feature extraction, followed by classifier training and evaluation. The only difference in our experimental setup is that we considered only half of the users from the MCYT and CEDAR datasets to ensure a fair comparison between their implementations

[4] https://github.com/dimTsourounis/FKD
https://github.com/tallesbrito/continual_sigver

Table 8
Comparison of existing works in WD settings evaluated with and without our method of prototypical signatures ($k = 50$). All experiments employed an SVM with an RBF kernel for verification, and the reported results show EER with a global threshold. For MCYT, training used 10 genuine signatures; for the other datasets, 12 were used. Prototypical signatures were obtained with a multi-dataset approach using CEDAR+MCYT+GPDS-S. The p-value corresponds to a paired t-test if normality (Shapiro–Wilk) was satisfied, or to a Wilcoxon signed-rank test otherwise. ROPE is defined as $[-0.015, 0.015]$. Results demonstrating statistical equivalence are highlighted.

Work	Dataset	Model	EER_{global}(%)		p-value	% in ROPE	95% HDI
			Replication	Proto (k=50)			
Tsourounis et al. [16]	CEDAR	ResNet18 CL + KD: GEOM & TCE	4.00 ± 1.23	4.88 ± 0.97	**0.17**	83.2	−0.005 – 0.023
		ResNet18 CL + KD: GEOM & BT	3.59 ± 0.83	3.98 ± 0.80	**0.39**	**97.9**	**−0.006 – 0.014**
		ResNet18 CL + KD: GEOM & BC	3.70 ± 0.58	4.27 ± 1.00	**0.22**	**95.6**	−0.005 – 0.016
	MCYT	ResNet18 CL + KD: GEOM & TCE	6.60 ± 0.65	6.58 ± 0.70	**0.95**	**99.9**	**−0.007 – 0.007**
		ResNet18 CL + KD: GEOM & BT	7.21 ± 0.80	6.74 ± 0.69	**0.11**	**99.7**	**−0.011 – 0.002**
		ResNet18 CL + KD: GEOM & BC	6.24 ± 0.79	5.84 ± 0.86	**0.36**	**98.3**	**−0.014 – 0.006**
Viana et al. [14]	CEDAR	Continual ResNet152	7.59 ± 0.89	7.16 ± 1.30	**0.45**	94.8	−0.017 – 0.009
		Continual SigNet	5.20 ± 0.81	5.62 ± 0.87	**0.28**	**98.6**	**−0.005 – 0.013**
		Continual ViT	7.61 ± 1.09	7.11 ± 0.90	**0.35**	94.4	−0.018 – 0.007
	MCYT	Continual ResNet152	7.36 ± 1.02	7.91 ± 0.92	**0.36**	91.8	−0.009 – 0.019
		Continual SigNet	5.88 ± 1.01	5.62 ± 1.02	**0.60**	**97.8**	**−0.014 – 0.009**
		Continual ViT	10.41 ± 1.25	11.32 ± 1.24	**0.11**	84.9	−0.003 – 0.022
	SGPDS150	Continual SigNet	5.76 ± 0.17	6.12 ± 0.40	0.03	**100.0**	**0.000 – 0.007**
		Continual ViT	7.21 ± 0.22	8.31 ± 0.42	0.00	**98.1**	**0.007 – 0.015**
	SGPDS300	Continual ResNet152	5.90 ± 0.23	6.17 ± 0.34	**0.10**	**100.0**	**−0.001 – 0.006**
		Continual SigNet	5.83 ± 0.23	6.04 ± 0.24	**0.06**	**100.0**	**−0.000 – 0.004**
		Continual ViT	7.72 ± 0.23	8.55 ± 0.26	0.00	**100.0**	**0.006 – 0.011**

Table 9
Comparison of existing works in WD settings evaluated with and without our method of prototypical signatures ($k = 50$). All experiments employed an SVM with an RBF kernel for verification, and the reported results show EER with user thresholds. For MCYT, training used 10 genuine signatures; for the other datasets, 12 were used. Prototypical signatures were obtained with a multi-dataset approach using CEDAR+MCYT+GPDS-S. The p-value corresponds to a paired t-test if normality (Shapiro–Wilk) was satisfied, or to a Wilcoxon signed-rank test otherwise. ROPE is defined as $[-0.015, 0.015]$. Results demonstrating statistical equivalence are highlighted.

Work	Dataset	Model	EER_{user}(%)		p-value	% in ROPE	95% HDI
			Replication	Proto (k=50)			
Tsourounis et al. [16]	CEDAR	ResNet18 CL + KD: GEOM & TCE	1.25 ± 0.35	1.46 ± 0.57	**0.46**	**99.9**	**−0.005 – 0.009**
		ResNet18 CL + KD: GEOM & BT	1.00 ± 0.38	1.09 ± 0.53	**0.65**	**100.0**	**−0.004 – 0.006**
		ResNet18 CL + KD: GEOM & BC	1.59 ± 0.43	1.62 ± 0.62	**0.89**	**100.0**	**−0.006 – 0.006**
	MCYT	ResNet18 CL + KD: GEOM & TCE	3.61 ± 0.72	4.11 ± 0.81	**0.37**	**98.0**	**−0.004 – 0.014**
		ResNet18 CL + KD: GEOM & BT	4.50 ± 0.83	3.77 ± 0.77	0.04	**98.1**	**−0.015 – 0.000**
		ResNet18 CL + KD: GEOM & BC	3.45 ± 1.02	3.88 ± 0.76	**0.35**	**97.9**	**−0.006 – 0.015**
Viana et al. [14]	CEDAR	Continual ResNet152	2.64 ± 0.67	2.71 ± 0.63	**0.81**	**99.9**	**−0.006 – 0.008**
		Continual SigNet	1.41 ± 0.65	2.48 ± 0.64	0.00	91.3	0.004 – 0.017
		Continual ViT	2.50 ± 0.70	3.30 ± 0.62	0.04	**96.1**	−0.000 – 0.016
	MCYT	Continual ResNet152	3.68 ± 1.04	4.68 ± 1.20	**0.17**	75.4	−0.007 – 0.026
		Continual SigNet	3.01 ± 0.75	3.02 ± 0.69	**0.98**	**99.7**	**−0.008 – 0.008**
		Continual ViT	5.68 ± 1.08	6.08 ± 1.43	**0.49**	92.8	−0.011 – 0.019
	SGPDS150	Continual SigNet	2.76 ± 0.28	3.68 ± 0.52	0.00	**99.4**	**0.005 – 0.013**
		Continual ViT	3.36 ± 0.47	3.95 ± 0.38	0.01	**99.9**	**0.001 – 0.011**
	SGPDS300	Continual ResNet152	2.62 ± 0.14	3.80 ± 0.23	0.00	**99.5**	**0.010 – 0.014**
		Continual SigNet	2.71 ± 0.20	3.73 ± 0.31	0.00	**99.6**	**0.007 – 0.013**
		Continual ViT	3.71 ± 0.31	4.31 ± 0.34	0.01	**100.0**	**0.003 – 0.009**

with and without the ProtoSig approach. This adjustment was necessary since the remaining half of the users were reserved for the summarization process, which was applied to the combined exploitation sets of MCYT, CEDAR, and GPDS-S (Table 2), using $k = 50$ prototypical signatures, as previously established in our validation experiments.

Results. Results for the writer-dependent approach are presented in Table 8, which considers EER with a global threshold, and in Table 9, which uses user-specific thresholds. As can be observed, the use of prototypical signatures yields results that are overall comparable to those obtained without our proposed method.

To assess whether the proposed prototypical training data generation technique leads to statistically equivalent performance, we conducted a systematic statistical analysis across datasets and models. For each paired configuration (with and without prototypes), we first evaluated the normality of the residuals using the Shapiro–Wilk test [37]. If the distribution of residuals did not significantly deviate from normality, we applied the paired Student's t-test [38]; otherwise, we employed the non-parametric Wilcoxon signed-rank test [39]. The corresponding results can be observed in the column "p-values".

As shown, the majority of comparisons yielded large p-values, suggesting that we could not reject the null hypothesis of equal distributions. To complement the frequentist analysis, considering it does not allow us to conclude equivalence [40], we performed a Bayesian inference using the highest density interval (HDI) with the Region of Practical Equivalence (ROPE) [41]. For this, we defined the ROPE as a small, negligible variation in EER $[-0.015, 0.015]$ and used the 95% HDI credible interval, as it is commonly used to characterize the uncertainty of the estimation [40].

Results reveal that, in the majority of cases, more than 95% of the posterior probability of the EER difference lies within the ROPE, indicating strong evidence for equivalence between approaches with and without our method. For example, in the experiments with ResNet18 backbones from [16], the posterior probability within the ROPE exceeded 97% in most datasets and configurations, with the 95% HDIs centered near zero. In the continual learning setting of [14], some p-values indicated significant differences (e.g., Continual SigNet on SGPDS150 and SGPDS300). However, in these cases, more than 99%

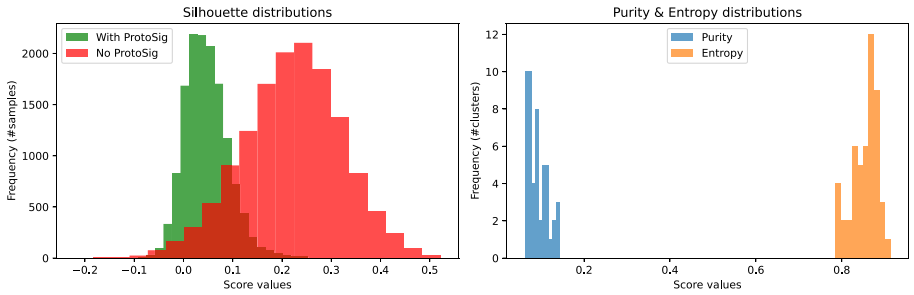


Fig. 13. Distributions of purity, entropy, and silhouette scores for GPDS-S ($k = 50$). Silhouette scores are shown with and without summarization.

of the posterior probability of the EER difference remained within the ROPE, with 95% HDIs fully contained in the predefined interval, confirming that the deviations are negligible in practice [40].

These findings demonstrate that the ProtoSig training strategy achieves performance that is statistically equivalent to the original approaches, while improving fairness and decreasing complexity. It is noteworthy that we did not fine-tune the systems to select the best hyperparameter k, but instead used a predefined $k = 50$. On top of that, we used the same combined dataset across all summarization processes, varying only the backbones. This highlights the generalization capability of our method and suggests that further gains may be attainable through backbone-specific optimization.

5.7. Data privacy

A growing concern in biometric verification is robustness to inverse biometrics attacks, where sensitive data can be reconstructed from stored templates [42,43]. We evaluated whether prototypical vectors may inadvertently leak sensitive information via two complementary analyses: (i) silhouette score [44] and (ii) cluster purity and entropy [45]. Complete mathematical definitions are detailed in Section 9 of the supplementary material.

The silhouette coefficient is commonly [46–48] employed in privacy-preserving works to evaluate how well clusters remain defined after privacy mechanisms are applied. In our context, the silhouette coefficient evaluates how well signatures belonging to the same writer remain grouped together compared to signatures from other writers after the summarization process. High silhouette values indicate that signatures from one writer are tightly clustered and far from those of other writers, suggesting that the resulting prototypical vectors remain strongly tied to specific users and could facilitate re-identification. Conversely, low or negative silhouette values occur when signatures from different writers overlap substantially, which dilutes writer-specific information and enhances privacy. To quantify the effect of our approach on silhouette scores, we compute the silhouette coefficient under two conditions: (i) without ProtoSig, where each user forms a cluster, and (ii) with ProtoSig, where the data is partitioned into k clusters.

Purity and entropy measure how strongly each cluster is associated with specific users. High purity indicates that most signatures in a cluster come from a single writer, suggesting that the corresponding prototypical vector may encode user-specific information. Conversely, high entropy indicates that a cluster aggregates signatures from many writers, thereby diluting personal information and enhancing privacy. To compute purity and entropy metrics, we first assign each signature embedding to its closest prototypical signature, then compute the proportion of samples belonging to the most frequent writer within each cluster (purity) and the entropy of the writer distribution inside the cluster. In terms of privacy, lower purity and higher entropy are desirable, as they indicate less dominance by any single writer and a more uniform distribution across users.

Fig. 13 presents results for GPDS-S ($k = 50$). Overall, the metrics support the privacy-preserving effect of our method. Total purity was 0.09, confirming no cluster is dominated by a single writer. Total entropy reached 0.85 (near the normalized maximum of 1), indicating clusters mix many users in nearly uniform proportions. The silhouette coefficient dropped from 0.22 (original feature vectors) to 0.04 after summarization, showing that signatures from different writers strongly overlap and are no longer well-separated in the feature space.

These results suggest that ProtoSig reduces identifiability compared to raw embeddings, which, in some cases, can be inverted to approximate the original biometric data [42,43]. The prototypical signatures correspond to cluster centroids mixing information from many individuals, providing practical privacy protection while preserving discriminative power.

6. Discussion

Experimental results demonstrate that ProtoSig provides a principled alternative to random-forgery sampling in writer-dependent handwritten signature verification, simultaneously enhancing stability, fairness, scalability, and privacy, properties rarely addressed jointly in existing HSV approaches.

The proposed paradigm also naturally extends to writer-independent verification, where prototypical signatures can be paired with genuine samples to form between-class dissimilarities, enabling WI training without reliance on random forgeries.

Regarding limitations, the experimental analysis focuses primarily on deep features, which dominate current HSV research [1]. Hence, the effectiveness of ProtoSig combined with handcrafted feature representations was not assessed in this study. Furthermore, the privacy evaluation relies on indirect, distribution-based measures and does not include explicit adversarial re-identification attacks, providing no formal privacy guarantees. Finally, ProtoSig does not explicitly target presentation attack detection (PAD) and should be viewed as complementary to PAD solutions.

7. Conclusion

This study introduced *ProtoSig*, a novel training data generation strategy for handwritten signature verification. The method applies a summarization process to the development data to generate prototypical signatures that serve as negative samples, enabling a deterministic selection that replaces arbitrary random forgery sampling. We conducted a comprehensive set of experiments covering verification performance, computational complexity, stability, fairness, privacy, and integration with existing systems.

ProtoSig offers several key advantages over traditional random sampling: (1) it generates more informative negative samples, achieving performance comparable to or better than the standard approach, even with fewer data; (2) it fosters fairer training through a more equitable distribution of negative samples across users; (3) it promotes greater stability and reproducibility by eliminating stochastic performance variability; (4) it improves scalability and reduces storage requirements; and (5) it reduces identifiability by relying on compact prototypical signatures aggregating information from multiple users.

When integrated into existing systems, ProtoSig achieves competitive performance even without fine-tuning, making it a practical solution for real-world deployments.

Future work will explore improved hyperparameter tuning, more adaptive learning of prototypical signatures, applicability to writer-independent settings, integration within PAD-aware pipelines, and cross-dataset evaluations to further assess generalization capability. Overall, this study opens a promising new line of research in HSV, highlighting that quality data matters more than quantity.

CRediT authorship contribution statement

Kecia G. de Moura: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization. **Robert Sabourin:** Writing – review & editing, Supervision, Resources, Methodology, Funding acquisition, Conceptualization. **Rafael M.O. Cruz:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization.

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.

Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), discovery grant program (RGPIN-2021-04130 and RGPIN-2022-02943). We would also like to thank the authors of [14,16] for releasing reproducible code and advancing research in the field.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patcog.2026.114258.

Data availability

The authors do not have permission to share data.

References

[1] A. Singla, A. Mittal, Exploring offline signature verification techniques: a survey based on methods and future directions, Multimedia Tools Appl. 84 (6) (2025) 2835–2875.

[2] M. Diaz, M.A. Ferrer, D. Impedovo, M.I. Malik, G. Pirlo, R. Plamondon, A perspective analysis of handwritten signature technology, ACM Comput. Surv. 51 (6) (2019).

[3] P.R. Prajapati, S. Poudel, M. Baduwal, S. Burlakoti, S.P. Panday, Signature verification using convolutional neural network and autoencoder, J. Inst. Eng. 16 (1) (2021) 33–40.

[4] D. Tsourounis, I. Theodorakopoulos, E.N. Zois, G. Economou, From text to signatures: Knowledge transfer for efficient deep feature learning in offline signature verification, Expert Syst. Appl. 189 (2022) 116136.

[5] T. Longjam, D.R. Kisku, P. Gupta, Writer independent handwritten signature verification on multi-scripted signatures using hybrid CNN-BiLSTM: A novel approach, Expert Syst. Appl. 214 (2023) 119111.

[6] D. Tsourounis, I. Theodorakopoulos, E.N. Zois, G. Economou, Leveraging expert models for training deep neural networks in scarce data domains: Application to offline handwritten signature verification, 2023, preprint arXiv:2308.01136.

[7] H. Li, P. Wei, Z. Ma, C. Li, N. Zheng, TransOSV: Offline signature verification with transformers, Pattern Recognit. 145 (2024) 109882.

[8] H. Zhang, J. Guo, K. Li, Y. Zhang, Y. Zhao, Offline signature verification based on feature disentangling aided variational autoencoder, in: Proceedings of the International Conference on Pattern Recognition, IEEE, 2024.

[9] N. Arab, H. Nemmour, M.L. Bouibed, Y. Chibani, 1D-GAN for improving offline handwritten signature verification based on small sets of real samples, Multimedia Tools Appl. 84 (21) (2025) 24541–24561.

[10] N. Arab, H. Nemmour, M.L. Bouibed, Y. Chibani, Handcrafted and deep features for synthetic data generation in offline handwritten signature verification, in: A. Bennour, A. Bouridane, S. Almaadeed, B. Bouaziz, E. Edirisinghe (Eds.), Intelligent Systems and Pattern Recognition, Springer Nature Switzerland, Cham, 2025, pp. 123–134.

[11] X. Zhao, L. Zheng, P. Yuan, Y. Zheng, Fusing deep and hand-crafted features by deep canonically correlated contractive autoencoder for offline signature verification, Pattern Recognit. 168 (2025) 111834.

[12] L. Zheng, D. Wu, S. Xu, Y. Zheng, HTCSigNet: A hybrid transformer and convolution signature network for offline signature verification, Pattern Recognit. 159 (2025) 111146.

[13] J.-X. Ren, Y.-J. Xiong, H. Zhan, B. Huang, 2C2S: A two-channel and two-stream transformer based framework for offline signature verification, Eng. Appl. Artif. Intell. 118 (2023) 105639.

[14] T.B. Viana, V.L.F. Souza, A.L.I. Oliveira, R.M.O. Cruz, R. Sabourin, Robust handwritten signature representation with continual learning of synthetic data over predefined real feature space, in: Document Analysis and Recognition - ICDAR 2024, Springer Nature Switzerland, Cham, 2024, pp. 233–249.

[15] Y.-J. Xiong, S.-Y. Cheng, J.-X. Ren, Y.-J. Zhang, Attention-based multiple siamese networks with primary representation guiding for offline signature verification, Int. J. Doc. Anal. Recognit. 27 (2) (2024).

[16] D. Tsourounis, I. Theodorakopoulos, E.N. Zois, G. Economou, A feature-based knowledge distillation (FKD) for offline signature feature learning without signatures, Expert Syst. Appl. (2025) 129158.

[17] S. Chattopadhyay, S. Manna, S. Bhattacharya, U. Pal, SURDS: Self-supervised attention-guided reconstruction and dual triplet loss for writer independent offline signature verification, in: 2022 26th International Conference on Pattern Recognition, ICPR, 2022, pp. 1600–1606.

[18] S. Manna, S. Chattopadhyay, S. Bhattacharya, U. Pal, SWIS: Self-supervised representation learning for writer independent offline signature verification, in: 2022 IEEE International Conference on Image Processing, ICIP, 2022.

[19] K.G. de Moura, R.M.O. Cruz, R. Sabourin, Offline handwritten signature verification using a stream-based approach, in: 2024 27th International Conference on Pattern Recognition, ICPR, 2024, pp. 271–286.

[20] E.N. Zois, M. Diaz, S. Said, M.A. Ferrer, Quasi-synthetic Riemannian data generation for writer-independent offline signature verification, 2025, arXiv:2509.20420.

[21] T.B. Viana, V.L. Souza, A.L. Oliveira, R.M. Cruz, R. Sabourin, A multi-task approach for contrastive learning of handwritten signature feature representations, Expert Syst. Appl. 217 (2023) 119589.

[22] N. Cohen, H.C. Indelman, Y. Fairstein, G. Kushilevitz, InDi: Informative and diverse sampling for dense retrieval, 2024.

[23] Y. Roh, K. Lee, S.E. Whang, C. Suh, Sample selection for fair and robust training, in: Neural Information Processing Systems, 2021.

[24] L.H. Clemmensen, R.D. Kjæ rsgaard, Data representativity for machine learning and AI systems, 2022, arXiv preprint arXiv:2203.04706.

[25] M.K. Kalera, S. Srihari, A. Xu, Offline signature verification and identification using distance statistics, Int. J. Pattern Recognit. Artif. Intell. 18 (07) (2004) 1339–1360.

[26] J. Ortega-Garcia, J. Fierrez, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernáez, J. Igarza, C. Vivaracho-Pascual, D. Escudero, Q. Moro-Sancho, MCYT baseline corpus: a bimodal biometric database, IEE Proc. - Vis. Image Signal Process. (2003) 395–401.

[27] M.A. Ferrer, M. Diaz, C. Carmona-Duarte, A. Morales, A behavioral handwriting model for static and dynamic signature synthesis, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1041–1053.

[28] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux, API design for machine learning software: experiences from the scikit-learn project, in: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108–122.

[29] E.N. Zois, I. Theodorakopoulos, D. Tsourounis, G. Economou, Parsimonious coding and verification of offline handwritten signatures, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017.

[30] R. Bolle, S. Pankanti, N. Ratha, Evaluation techniques for biometrics-based authentication systems (FRR), in: Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, vol. 2, 2000, pp. 831–837 vol.2.

[31] H. Jagode, A. Danalis, G. Congiu, D. Barry, A. Castaldo, J. Dongarra, Advancements of PAPI for the exascale generation, Int. J. High Perform. Comput. Appl. 39 (2) (2025) 251–268.

[32] C. Garcia-Cardona, B. Wohlberg, Convolutional dictionary learning: A comparative review and new algorithms, IEEE Trans. Comput. Imaging 4 (3) (2018) 366–381.

[33] A.K. Jain, D. Deb, J.J. Engelsma, Biometrics: Trust, but verify, IEEE Trans. Biom. Behav. Identity Sci. 4 (3) (2022) 303–323.

[34] M.O. Lorenz, Methods of measuring the concentration of wealth, Publ. Am. Stat. Assoc. 9 (70) (1905) 209–219, (Accessed 2026-01-29).
[35] C. Gini, Variabilità e mutabilità, 1912.
[36] S. Eberz, K.B. Rasmussen, V. Lenders, I. Martinovic, Evaluating behavioral biometrics for continuous authentication: Challenges and metrics, in: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, in: ASIA CCS '17, Association for Computing Machinery, 2017, pp. 386–399.
[37] S.S. Shapiro, M.B. Wilk, An analysis of variance test for normality (complete samples)†, Biometrika 52 (3–4) (1965) 591–611.
[38] Student, The probable error of a mean, Biometrika 6 (1) (1908) 1–25, Full publication date: Mar., 1908.
[39] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics 1 (1945) 196–202.
[40] J. Pan, R. Christoforou, L.N. Wiedermann, M. Schweiker, The untapped potential of Bayesian region of practical equivalence for assessing null effects in multi-domain research, Build. Environ. 283 (2025) 113390.
[41] J.K. Kruschke, T.M. Liddell, The Bayesian new statistics: Hypothesis testing, estimation, meta-analysis, and power analysis from a Bayesian perspective, Psychon. Bull. Rev. 25 (1) (2018) 178–206.
[42] S.M. Abdullahi, S. Sun, B. Wang, N. Wei, H. Wang, Biometric template attacks and recent protection mechanisms: A survey, Inf. Fusion 103 (2024).
[43] M. Gomez-Barrero, J. Galbally, Reversing the irreversible: A survey on inverse biometrics, Comput. Secur. 90 (2020) 101700.
[44] P.J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.
[45] P.-N. Tan, M. Steinbach, A. Karpatne, V. Kumar, Introduction to Data Mining, Pearson, 2018.
[46] M.E. Ferrão, P. Prata, P. Fazendeiro, Utility-driven assessment of anonymized data via clustering, Sci. Data 9 (1) (2022) 456.
[47] K. Sathiyapriya, R.K. Aravindhan, B. Kireshvanth, Y. Ranganathan, P. Hardik, Privacy-preserving social network clustering using differential privacy, in: 2024 ICSSEECC, 2024, pp. 398–403.
[48] M. Ghasemian, E. Ayday, Privacy-preserving optimal parameter selection for collaborative clustering, in: Computer Security. ESORICS 2024 International Workshops, Springer Nature Switzerland, Cham, 2025, pp. 13–32.