Pattern Recognition Letters xxx (xxxx) xxx

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Offline handwritten signature verification under a stream context with partially labeled data

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

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

ARTICLE INFO

Editor: Maria De Marsico

Keywords:
Offline signature
Handwritten signature
Data stream
Dissimilarity data
Active learning
Adaptive classifier

ABSTRACT

Handwritten signature verification is a biometric authentication task that distinguishes genuine from forged signatures. Traditional offline methods rely on batch learning, which struggles with signature variability and limited labeled data. Recent advances in stream-based approaches have improved adaptability but still depend on fully labeled training data, which is impractical for real-world applications. This paper extends prior work by introducing a framework that integrates partially labeled data with active learning to enhance verification performance in a streaming context. We introduce a novel query strategy based on the k-disagreeing neighbors score, which prioritizes ambiguous samples near decision boundaries. Experimental results demonstrate that the proposed approach adapts to new signature variations while improving performance with minimal labeled data. Random sampling proves surprisingly effective by leveraging the structured dissimilarity space, while the k-disagreeing neighbors strategy provides more stable performance over time. The implementation is available at https://github.com/kdmoura/stream_hsv.

1. Introduction

Handwritten Signature Verification (HSV) is a biometric authentication task that aims to distinguish between genuine and forged signatures. HSV systems are broadly categorized into online and offline approaches [1]. Offline HSV systems deal with signature images obtained after the signing process is complete, whereas online HSV focuses on the writing dynamics [1].

Traditional offline HSV approaches [2,3] typically rely on batch learning paradigms, in which a static dataset is used to train a classifier before deployment. These models are expected to generalize across all enrolled users and their signatures without further updates, which would otherwise require retraining the entire model on both old and new data, demanding substantial time and indefinite storage capacity.

Deploying an HSV system without further training poses risks in the verification process as signature patterns evolve over time [1]. Real-world applications often face significant challenges due to the inherent variability of handwritten signatures, where genuine signatures from the same individual may exhibit natural variations caused by factors such as mood, aging, or writing conditions [1]. Additionally, the limited availability of labeled data poses another challenge, as collecting annotated signatures is expensive and labor-intensive [4]. Static models may fail to detect new forgeries or adapt to unseen variations, leading to increased false acceptances or rejections as time progresses. This lack of adaptability compromises the system's long-term reliability. This effect is conceptually illustrated in Fig. 1.

In our previous work [5], we introduced a stream-based approach for HSV, in which signatures are processed sequentially, allowing the system to continuously update its knowledge. The approach relies on a stream dichotomy transformation mechanism that reduces the multi-class biometric problem to a binary problem by leveraging differences between the claimed and reference signatures [6]. This transformation produces pairwise comparisons that map the problem to a dissimilarity space, where each comparison represents the degree of dissimilarity between signatures. Operating within the dissimilarity space enables the implementation of a writer-independent (WI) system, which creates a single model for all users. This eliminates the need to train and maintain separate classifiers for each user, as required in the writer-dependent (WD) approach [7], thereby significantly reducing computational costs.

While the stream HSV improved adaptability, it relied on fully labeled training data, which is often impractical in real-world applications. In classification under stream contexts with a scarcity of labels, there are two scenarios: unlabeled settings and partially labeled streams [4]. In the unlabeled setting, the true class label of an incoming instance

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

https://doi.org/10.1016/j.patrec.2026.05.018
Received 5 April 2025; Received in revised form 27 April 2026; Accepted 24 May 2026
Available online 28 May 2026
0167-8655/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

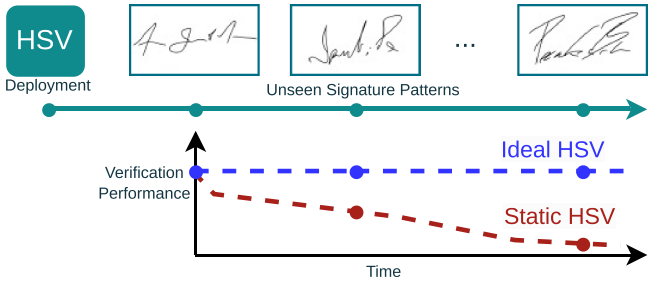


Fig. 1. Conceptual illustration of HSV performance over time without model updates. As new signatures arrive, a static model may degrade due to evolving patterns.

is not immediately available, or may arrive with such significant delay that makes it unusable for timely model updates [4]. In the partially labeled setting, some labels are available as the stream progresses, or there is an associated labeling budget that allows labels to be requested from an expert within a timeframe suitable for updating the classifier [4].

One common approach for partially labeled streams is Active Learning (AL) [4]. Unlike passive learning, in which models are trained on any available labels, AL allows the model to actively select which instances to annotate. This is achieved through a query strategy, which is a criterion for scoring unlabeled instances by their expected contribution to improving the model and a budget that limits the total number of requests [8]. Query strategies can be *information-based*, which favor instances near the decision boundary where the model is most uncertain (e.g., uncertainty sampling, margin sampling), or *representation-based*, which favor instances that are broadly representative of the input space (e.g., density-based or diversity-based sampling) [8]. The AL cycle works as follows: incoming unlabeled instances are evaluated, the most valuable ones are selected and submitted to a human annotator (oracle), and the resulting labels are used to update the model. By focusing on the most informative or ambiguous signatures rather than labeling data indiscriminately, active learning improves learning while reducing annotation costs [8].

In this work, we propose incorporating *partially labeled data* with *active learning* to further explore the efficiency of HSV systems in a stream context. Partially labeled data scenarios naturally arise in biometric applications, where signatures are continuously received and must be processed sequentially, often with limited annotation resources. One example of such a scenario is observed in vote-by-mail elections, where millions of voters sign ballot envelopes that must be verified before they are counted [9]. Due to the high volume of incoming signatures [10], it is impractical to manually inspect every instance, making it crucial to apply techniques such as active learning to prioritize the most informative cases for annotation.

To address this challenge, we explore different query strategies for active learning, including random sampling, hyperplane distance [2], and a novel approach based on the k-Disagreeing Neighbors (kDN) score [11], which quantifies the proportion of nearest neighbors with differing labels to provide insight into decision space complexity. We validated our approach through experiments on the GPDS Synthetic, CEDAR, and MCYT-75 datasets, in which the system processes incoming sequential signatures and is updated using active learning strategies. Evaluation under varying labeling budgets and initial training set sizes demonstrates that the system maintains high verification performance with minimal labeled data, even with limited initial training. Notably, random sampling achieved competitive performance, while k-Disagreeing Neighbors provided more stable performance over time, highlighting the potential of simple yet robust query strategies in dissimilarity-based stream handwritten signature verification.

This paper is an extended version of our recent conference paper [5]. The extensions are as follows:

- We expanded the dissimilarity-based stream HSV framework to incorporate partially labeled data, enabling the system to operate even when only a fraction of the data is annotated.
- We integrate active learning strategies to optimize the selection of labeled samples, reducing annotation costs while maintaining high verification performance.
- We introduce and evaluate the k-Disagreeing Neighbors(kDN) score as a novel query strategy for active learning in handwritten signature verification.

2. Related work

Works on offline HSV have predominantly used batch learning approaches, where models are trained on a static dataset before deployment [7]. These models remain unchanged during operation, limiting their ability to adapt to the inherent variations in signature patterns.

Under a stream context, many approaches have been proposed for continuous authentication using different biometrics traits such as keystroke dynamics [12–14], mouse movements [12,14], application usage [12], the system footprint [12], face and fingerprint [15] or also by applying multimodal approaches, employing a combination of biometrics variables with fusing strategies [15,16]. Our previous work [5] introduced the first stream-based approach for offline handwritten signatures, addressing their intrinsic variability and evolving nature. Nonetheless, the proposal assumed the availability of labels for all arriving samples, which may be impractical in real-world applications.

In the context of partially labeled data for offline handwritten signature verification, human collaboration can be a viable strategy for obtaining labels. Morales et al. [17] analyzed how human intervention can contribute to signature authentication at classification and feature extraction levels. Their crowdsourcing experiments with 500 laypeople established a human baseline, with individual error rates around 30%, whereas aggregated opinions substantially reduced errors. At the feature level, manual annotation of signature attributes yielded improvements of 16 to 90% when combined with HSV systems, depending on the scenario. However, exhaustive human annotation remains costly and impractical at scale, motivating the use of AL to selectively query only the most informative samples.

Younesian et al. [2] introduced the only active learning-based approach for HSV. They apply AL in the classification step, using a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel to perform writer-dependent verification. Three query strategies were tested: (1) Distance-based sampling (selecting instances near the SVM hyperplane), (2) Maximum Entropy (choosing instances with the highest label uncertainty), and (3) K-Nearest Neighbors (selecting the most distant instance among K-neighbors). Although their method improved performance by nearly 13% over random sampling, it was evaluated solely on Persian signatures from the UTSig dataset and followed a writer-dependent approach under batch settings.

Writer-dependent models require training and maintaining separate classifiers for each user, making them computationally expensive and impractical for real-world scenarios with continuously arriving data. In contrast, our approach employs a writer-independent framework, enabling a single model to generalize across users while incorporating active learning in a stream context to efficiently handle partially labeled data. To the best of our knowledge, no prior work has investigated offline signature verification in streaming contexts or explored the dissimilarity space in such a dynamic scenario. Furthermore, this work is the first to propose kDN-based active learning for query selection in signature verification.

3. Dissimilarity-based stream HSV with active learning

Our proposed system is illustrated in Fig. 2. The system takes a sequence of raw signature images as input, denoted by Stream =

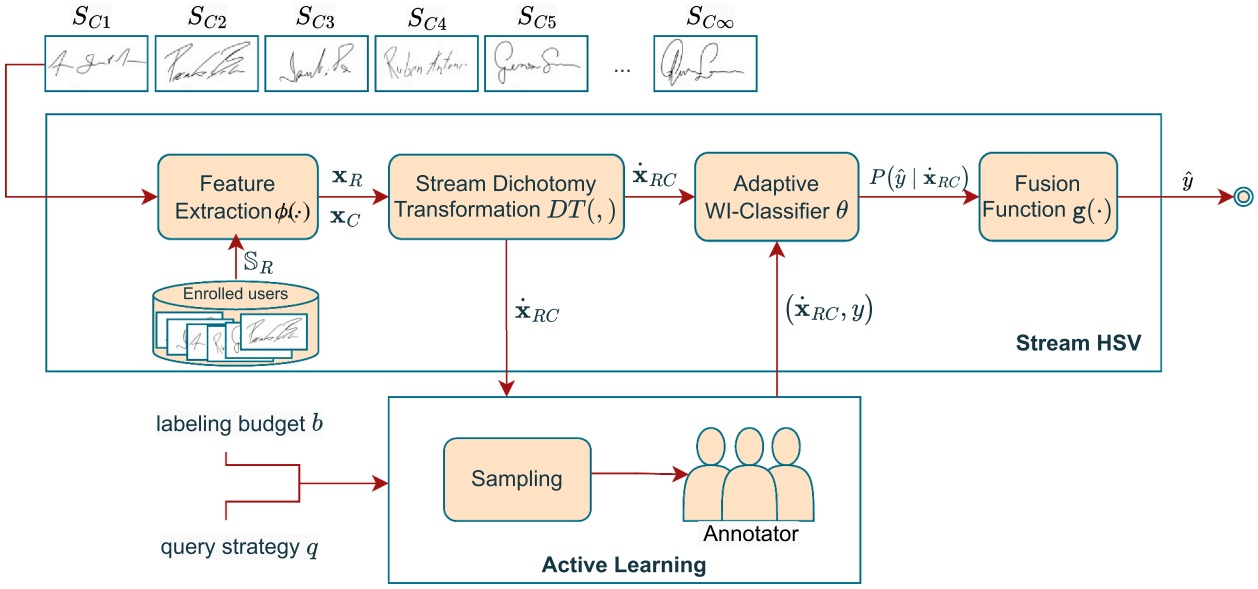


Fig. 2. Dissimilarity-based stream HSV with active learning. A stream of claimed signature images $S_{C1}, S_{C2}, \ldots, S_{C\infty}$ is preprocessed and transformed into feature vectors using a representation model ϕ. The stream dichotomy transformation is applied to a pair of reference and claimed feature vectors, which are then classified by an adaptive classifier θ. A fusion function aggregates predictions across all reference signatures to produce a final result. Tested signatures are sampled under a query strategy q within a labeling budget b, and selected samples with respective true labels y are used to update the classifier.

$\{S_{C1}, S_{C2}, \ldots, S_{C\infty}\}$, where S_{Ci} represents the ith claimed incoming signature image. The system verifies these images against enrolled user reference samples, generating dissimilarity vectors with the stream dichotomy transformation DT. An adaptive writer-independent classifier θ performs the verification by processing dissimilarity representations. Since the stream data is partially labeled, an active learning query strategy q selects the most informative dissimilarity vectors $\dot{\mathbf{x}}_{RC}$ for labeling within a labeling budget b. These samples are subsequently used to update the classifier θ. This iterative process enables the system to gradually improve performance while minimizing annotation effort.

Representation Model $\phi(\cdot)$. The representation model ϕ converts the raw signature image input into a compact feature vector by extracting meaningful and discriminative features. Given a signature image S_C from a claimed user, the extracted feature vector is expressed as $\mathbf{x}_C = \phi(S_C)$. Pretrained or custom-trained models can be used to optimize feature extraction.

Stream Dichotomy Transformation $DT(,)$. In the proposed framework, the dichotomy transformation is applied to each incoming signature after the feature vector is extracted. This is performed by comparing the input vector against each reference sample of the claimed user. The process is defined as the absolute distance between two feature vectors. That is, given a vector $\mathbf{x}_C$ (claimed vector) and $\mathbf{x}_R$ (reference vector):

$$DT(\mathbf{x}_C, \mathbf{x}_R) = \begin{bmatrix} |x_{C1} - x_{R1}| \\ |x_{C2} - x_{R2}| \\ \vdots \\ |x_{Cn} - x_{Rn}| \end{bmatrix} = \dot{\mathbf{x}}_{RC}$$

If $\dot{\mathbf{x}}_{RC}$ is obtained from signatures of the same user, it is labeled as *positive*. Otherwise, it is labeled as *negative*.

The $DT(,)$ method converts multi-class problems into binary classification tasks [18], enabling a writer-independent framework (i.e., a unified classifier) which is essential for the stream context. Additionally, it mitigates the class imbalance issue in data streams [19], allowing the generation of an equal number of negative samples for each positive input.

Adaptive WI-Classifier θ. Unlike static HSV systems, which often use SVMs with RBF kernel [20], our work requires an adaptive classifier. We adopt a Stochastic Gradient Descent (SGD) classifier with a hinge loss function, following prior works [21–24] that minimizes the loss function of a linear SVM in the primal formulation directly. This approach avoids the computational overhead of calculating and storing the dual variables as required in the dual formulation. The loss function (Eq. (1)) balances margin maximization and misclassification penalties using the regularization parameter C.

$$\min_{\mathbf{w},b} \quad \frac{1}{N}\sum_{i=1}^{N} \max(0, 1 - y_i(\mathbf{w} \cdot \mathbf{x}_i + b)) + \frac{C}{2}||\mathbf{w}||^2 \tag{1}$$

The updating process of the WI-classifier relies on label data, obtained via active learning approaches discussed in the following section. The WI-classifier's output is based on the distance between dissimilarity vectors and its decision hyperplane. Let $\dot{\mathbf{x}}_1, \dot{\mathbf{x}}_2, \ldots, \dot{\mathbf{x}}_m$ be the dissimilarity vectors of a claimed user signature and their m reference samples. The classifier θ returns the signed distance $d_i = \theta(\dot{\mathbf{x}}_i)$, where $i = 1, \ldots, m$. These distances are combined by a fusion function $g(\cdot)$. We adopt the Max fusion function due to its superior performance, as demonstrated by [6], where $g = \max(d_1, d_2, \ldots, d_m)$.

Active Learning. The goal of integrating active learning into the system is to achieve performance comparable to fully labeled systems while reducing the number of annotated instances. A labeling budget $b \in [0, 1]$ controls the proportion of labeled data, where $b = 1$ means all instances are labeled, and $b = 0$ means no additional annotation (i.e., no updates).

At each time step, a sequence of signatures arrives for verification. Given a chunk ch_{size} of arriving data $\mathbb{S}_{\text{chunk}} = \{S_1, S_2, \ldots, S_{ch_{size}}\}$, and m stored reference samples, the DT computes the dissimilarities $\dot{\mathbb{X}}_j = \{\dot{\mathbf{x}}_1, \dot{\mathbf{x}}_2, \ldots, \dot{\mathbf{x}}_m\}_j$, with $j = 1, \ldots, ch_{size}$. The resulting chunk of dissimilarities $\dot{\mathbb{X}}_{\text{chunk}} = \{\dot{\mathbb{X}}_1, \dot{\mathbb{X}}_2, \ldots, \dot{\mathbb{X}}_{ch_{size}}\}$ is passed to the active learning component. The AL method selects $b \times ch_{size}$ signature images from $\mathbb{S}_{\text{chunk}}$ for labeling based on a query strategy q. We investigate different query strategies described below.

3.1. kDN sampling

In this work, we propose a novel active learning strategy based on the K-Disagreeing Neighbors score (kDN) [11] measure for sample selection. The kDN metric quantifies the proportion of neighbors with different labels in a sample's neighborhood, which provides insight into the complexity of the decision space [25]. In our work, the kDN is computed as defined in Eq. (2):

$$\mathrm{kDN}(\dot{\mathbf{x}}_q) = \frac{|\{\dot{\mathbf{x}}_k \in \mathrm{KNN}(\dot{\mathbf{x}}_q) \mid \mathrm{label}(\dot{\mathbf{x}}_k) \neq \mathrm{label}(\dot{\mathbf{x}}_q)\}|}{k} \tag{2}$$

where $\mathrm{KNN}(\dot{\mathbf{x}}_q)$ represents the k nearest neighbors of $\dot{\mathbf{x}}_q$ considering the initial training data, and $\mathrm{label}(\dot{\mathbf{x}})$ denotes the class label of sample $\dot{\mathbf{x}}$.

Based on the dissimilarity space distribution [6], we hypothesize that selecting samples with higher kDN values prioritizes ambiguous instances located in overlapping regions of the dissimilarity space, where the decision boundary is less distinct. Considering a chunk of signatures $\mathbb{S}_{\mathrm{chunk}}$, its corresponding dissimilarity vectors $\dot{\mathbb{X}}_{\mathrm{chunk}}$, and m reference signatures, the query set $\mathcal{Q}$ is obtained according to Eq. (3). First, we compute the maximum kDN value within each subset $\dot{\mathbb{X}}_j \in \dot{\mathbb{X}}_{\mathrm{chunk}}$, with $j = 1, \ldots, ch_{size}$. The resulting vector of maximum kDN values is sorted in descending order. Finally, the indices corresponding to the top h values (highest kDN) are selected to form $\mathcal{Q}$.

$$\mathcal{Q} = \arg\mathrm{sort}_{\mathrm{desc},h}\left[\left(\max \mathrm{kDN}(\dot{\mathbb{X}}_j)\right)_{j=1}^{ch_{size}}\right] \tag{3}$$

where $\dot{\mathbb{X}}_j = \{\dot{\mathbf{x}}_1, \dot{\mathbf{x}}_2, \ldots, \dot{\mathbf{x}}_m\}_j$ and $|\mathcal{Q}| = h = b \times ch_{\mathrm{size}}$. We adopt $k = 7$ as [26] indicates that at least 5 out of 7 nearest neighbors of a genuine sample belong to the same class, ensuring a reliable measure of local consistency, while increasing k further leads to more significant class overlap, potentially reducing the effectiveness of kDN estimation.

3.2. Random sampling

Random sampling selects a subset $\mathcal{Q}$ of signatures uniformly at random without considering any underlying structure or classifier information. This strategy is included as a baseline to provide a reference point for evaluating more advanced selection methods. The query set $\mathcal{Q}$ is sampled following Eq. (4), where h signatures are randomly selected from $\mathbb{S}_{\mathrm{chunk}}$:

$$\mathcal{Q} \sim \mathrm{Uniform}(\mathbb{S}_{\mathrm{chunk}}, h), \quad |\mathcal{Q}| = h = b \times ch_{\mathrm{size}} \tag{4}$$

3.3. Hyperplane distance

Based on [2], instances closest to the decision boundary of the WI-classifier are selected, as they represent the most uncertain samples. Let $d(\dot{\mathbf{x}})$ be the signed distance determined by the classifier to a dissimilarity vector $\dot{\mathbf{x}}$. Given $\mathbb{S}_{\mathrm{chunk}}$, $\dot{\mathbb{X}}_{\mathrm{chunk}}$, and m reference signatures, the query set $\mathcal{Q}$ is calculated via Eq. (5). First, the minimum absolute distance $|d|$ within each subset $\dot{\mathbb{X}}_j = \{\dot{\mathbf{x}}_1, \dot{\mathbf{x}}_2, \ldots, \dot{\mathbf{x}}_m\}$ is computed, with $\dot{\mathbb{X}}_j \in \dot{\mathbb{X}}_{\mathrm{chunk}}$ where $j = 1, \ldots, ch_{size}$. The resulting vector of minimum distances is then sorted in ascending order. Lastly, the indices corresponding to the top h values (smallest absolute distances) are selected to produce $\mathcal{Q}$.

$$\mathcal{Q} = \arg\mathrm{sort}_{\mathrm{asc},h}\left[\left(\min |d(\dot{\mathbb{X}}_j)|\right)_{j=1}^{ch_{size}}\right], \quad |\mathcal{Q}| = h = b \times ch_{\mathrm{size}} \tag{5}$$

4. Experimental setup

Datasets. We utilize the datasets described in Table 1. Following the process for stream evaluation described in [5], datasets are split into two disjoint sets: the development $\mathcal{D}$, used for pre-training the system, and the exploitation set $\mathcal{E}$, utilized for creating the stream of signatures employed for testing and updating the model.

Preprocessing. Before extracting features, signature images are first centered on a canvas sized to the dataset's largest sample. Otsu's algorithm removes the background, setting it to white and the foreground to grayscale. The image is then inverted, resized to 170×242 pixels, and center-cropped to 150×220 pixels.

Table 1
Summary of the datasets used in this work.

Dataset Name	Language	Users	Genuine signatures	Forgeries
CEDAR [27]	Western	55	24	24
GPDS Synthetic [28]	Western	10000	24	30
MCYT-75 [29]	Western	75	15	15

Table 2
Exploitation set ($\mathcal{E}$).

Stream	Data	Users	Reference sig.	Claimed sig.	
			m	t	set
1	GPDS-S	300	12	10	10G, 10RF, 10SK
2	CEDAR	55	10	10	10G, 10RF, 10SK
	MCYT	75	10	5	5G, 5RF, 5SK

Representation Model. We use SigNet Synthetic (*SigNet-S*) [3], a deep convolutional neural network trained on synthetic GPDS data. *SigNet-S* was developed to learn writer-independent signature features, focusing on discriminative attributes for individual writers. To extract features, the model processes the signature images via feedforward propagation, producing a 2048-dimensional feature vector.

Development Sets $\mathcal{D}$. After preprocessing and feature extraction, dissimilarity samples are generated. A subset of users from $\mathcal{D}$ is selected, specifically $\#\mathrm{Users} = \{5, 10, 50, 581\}$, where 581 represents the entire dataset. For each user in the selected subset, 12 genuine samples and 11 random forgery samples (genuine signatures from other users) are randomly selected. The dissimilarity training data is then constructed using the dichotomy transformation: within-class comparisons among all genuine samples form the positive class. At the same time, comparisons between eleven genuine and all six random forgeries form the negative class. This process is repeated for each subset configuration, yielding four distinct pre-training datasets. We adopt the configuration from [5] with the highest number of samples, as the behavior with fewer signatures remains similar.

Exploitation Sets $\mathcal{E}$. Considering each configuration shown in Table 2, for each user in $\mathcal{E}$, after preprocessing and feature extraction, m reference (genuine) signatures are randomly selected. Similarly, for each user, subsets of t genuine signatures, t random forgeries, and t skilled forgeries (forged signatures of the same user) are randomly chosen. Consequently, the exploitation set comprises four signature groups for each user: reference signatures ($\mathbb{S}_R$), genuine signatures ($\mathbb{S}_G$), random forgeries ($\mathbb{S}_{RF}$), and skilled forgeries ($\mathbb{S}_{SK}$), as illustrated in Fig. 3.

Stream Generation. As in [5], the stream is generated from the exploitation set, as illustrated in Fig. 3. To construct the stream of signatures, the samples in $\mathcal{E}$ are divided into blocks, each containing signatures from all users. For each user, three claimed signatures-a genuine signature (G), a random forgery (RF), and a skilled forgery (SK)-are selected for verification. These signatures are compared against the reference set to compute dissimilarity samples, which are then passed to the classifier.

Positive dissimilarity samples are generated by comparing reference signatures with genuine signatures. Negative samples are created in two ways: by comparing the reference set with random forgeries and with skilled forgeries.

Model Update with AL. Model is updated using active learning, with a labeling budget $b = \{0.01, 0.05, 0.1, 1.0\}$, where 1.0 represents a fully supervised configuration with 100% of arriving samples labeled. Only genuine and random forgeries are used in the update step.

Model Evaluation. We employ the interleaved test-then-train (prequential) evaluation approach, where the model is first tested on each

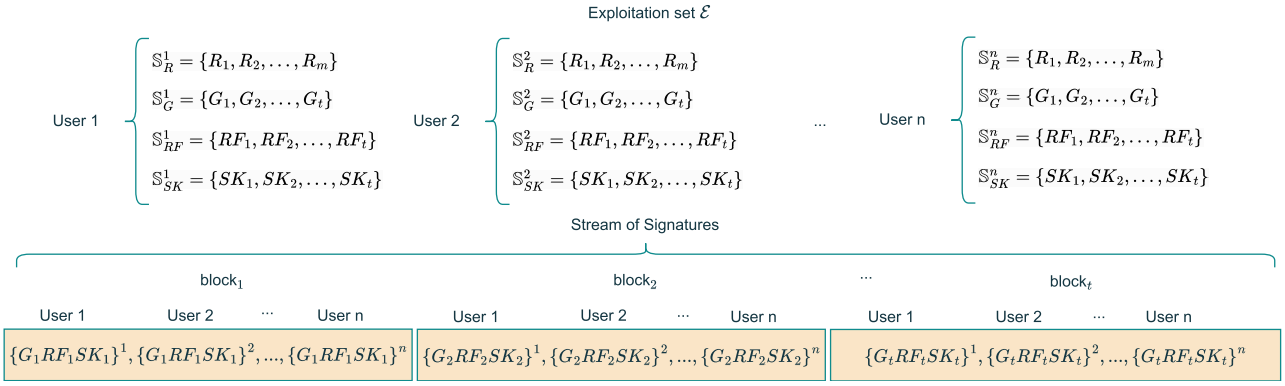


Fig. 3. Stream of arriving signatures derived from the exploitation set $\mathcal{E}$. $\mathbb{S}_R$, $\mathbb{S}_G$, $\mathbb{S}_{RF}$, and $\mathbb{S}_{SK}$ represent the randomly selected sets of reference, genuine, random forgery, and skilled forgery signatures, respectively. m denotes the number of reference signatures, while n and t correspond to the number of users and the number of claimed signatures, respectively.

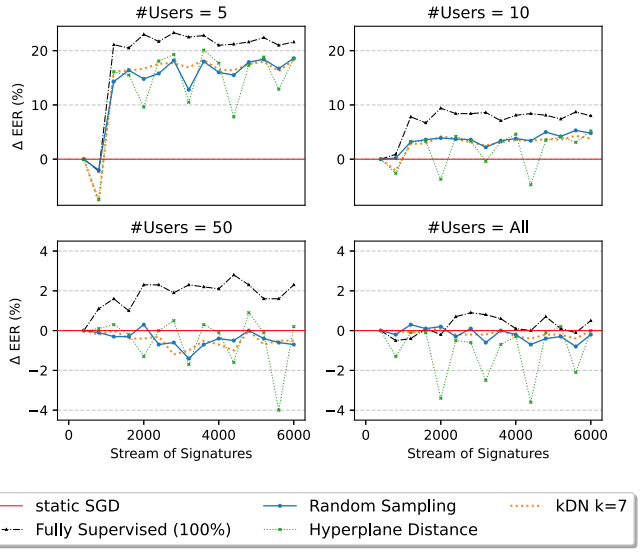


Fig. 4. Performance comparison between stream HSV and static HSV for skilled forgery detection across active learning strategies on the GPDS synthetic stream. The x-axis represents the cumulative number of processed signatures, while the y-axis shows the change in EER relative to the static model. Values above zero indicate an improvement (gain) over the static approach, while values below zero indicate a degradation (loss). AL labeling budget: 1%. #User denotes the number of users in the initial training data. Model updates at each chunk with $ch_{size} = 600$ samples, while evaluations are conducted over a sliding window with $w_{size} = w_{step} = 400$ samples.

incoming chunk of samples before being updated [30]. For evaluation, we use a window size (w_{size}), which represents the number of most recently tested signatures used for metric calculation, and a window step (w_{step}), which indicates how often the metrics are updated. After every w_{step} of signatures, the last w_{size} signatures are used to compute performance measures. We compute the Equal Error Rate (EER) using a global threshold. Each experiment is repeated five times, and we report the average results along with the standard deviation.

5. Experimental results

Fig. 4 illustrates the difference in EER of stream HSV compared to static HSV across different active learning query strategies for the GPDS Synthetic stream. The results highlight the system's ability to surpass the static approach even with a minimal labeling budget of 1% when initially trained on a limited number of users (#Users $\leq$ 10). As the initial training set size increases (#Users $\geq$ 50), stream HSV achieves performance comparable to the static HSV system with select active learning strategies while requiring significantly fewer labeled samples. These findings extend the results in [5], demonstrating that stream HSV not only reduces the required size of the initial training set but also minimizes the number of labeled samples needed to maintain efficient performance over time. A more comprehensive set of results, including extended labeling budget analysis and performance trends across all streams and strategies, are provided in the supplementary material.

When evaluating the impact of different query strategies, we observe that kDN sampling and random sampling produce similar results. Nevertheless, kDN shows more stable performance with fewer variations over time across all configurations. In contrast, the hyperplane-based strategy struggles to consistently select samples that improve performance as time progresses. This suggests that relying solely on the model's predictions does not guarantee performance gains, as the hyperplane distance may misrepresent uncertainty, leading to the selection of noisy or ineffective samples, causing performance to constantly shift.

Fig. 5 presents the impact of different labeling budgets using random sampling and kDN score methods. Due to space constraints, results for the hyperplane distance approach were omitted, as it exhibits a similar behavior and can be found in the supplementary material. Overall, the findings indicate that higher labeling budgets generally yield better performance across most configurations, regardless of the query strategy employed.

Nonetheless, a counterintuitive behavior emerges in the kDN score approach when the initial training set comprises a larger number of users, providing more samples for pre-training. In such a scenario, selecting only 1% of the data delivers better long-term performance compared to a 10% labeling budget. In other words, when the system is already well-trained, narrowing the selection to the samples with the highest kDN scores has a greater impact on refining decision boundaries.

To assess results on other datasets, Fig. 6 illustrates the impact of stream HSV on EER compared to static HSV when incoming signatures belong to a different distribution than the one used during initial training. In this setup, signatures arrive randomly from CEDAR or MCYT. Even with a small labeling budget of just 1%, performance improves over time, eventually approaching that of the fully supervised approach. This highlights the generalization capability of the dissimilarity-based stream method in handling signatures from unknown distributions, even with minimal labeled samples and under limited initial data.

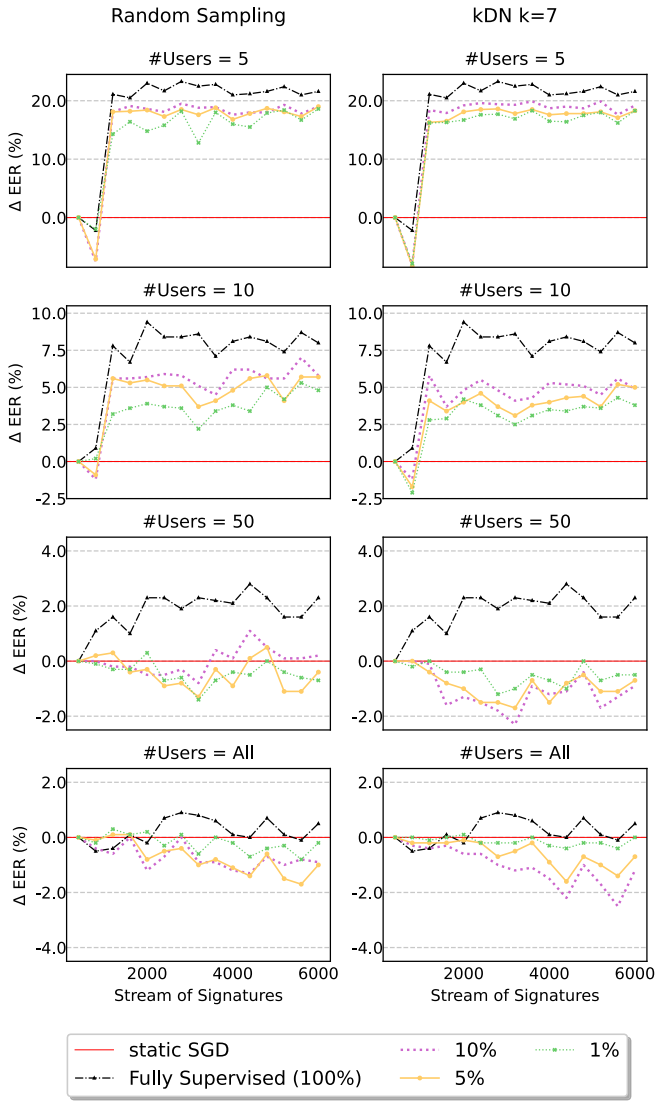


Fig. 5. Performance comparison between stream HSV and static HSV for skilled forgery detection across AL labeling budgets on the GPDS synthetic stream. The x-axis represents the cumulative number of processed signatures, while the y-axis shows the change in EER relative to the static model. Values above zero indicate an improvement (gain) over the static approach, while values below zero indicate a degradation (loss). Model updates at each chunk with $ch_{size} = 600$ samples, while evaluations are conducted over a sliding window with $w_{size} = w_{step} = 400$ samples.

Fig. 6 also shows that all query strategies proved effective in selecting samples, with the hyperplane distance-based approach yielding the best overall results, followed by random and kDN sampling. However, the results of the hyperplane method are inconsistent with those found for the GPDS Synthetic stream, suggesting sensitivity to data characteristics or evaluation parameters. In contrast, despite its simplicity, random sampling consistently improved system performance across different configurations.

To assess whether performance differences among models are statistically significant, we conducted a Friedman test at significance level $\alpha = 0.05$ [31]. The Friedman test ranks each method by EER across both evaluated streams (GPDS-S and CEDAR + MCYT) and initial training conditions (#Users = 5, 10, 50, All).

Two evaluation scenarios were considered: average prequential results over the stream and the most recent sliding window results [32]. For the former, the null hypothesis of equivalent performance was not

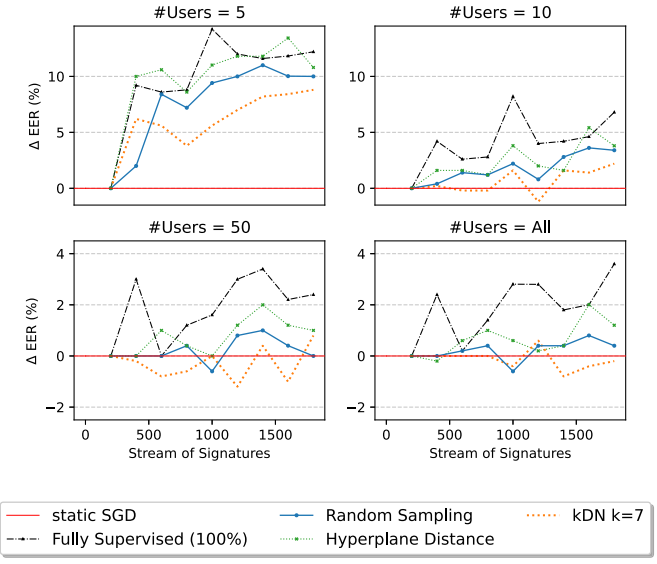


Fig. 6. Performance comparison between stream HSV and static HSV for skilled forgery detection across active learning strategies on the CEDAR + MCYT stream. The x-axis represents the cumulative number of processed signatures, while the y-axis shows the change in EER relative to the static model. Values above zero indicate an improvement (gain) over the static approach, while values below zero indicate a degradation (loss). AL labeling budget: 1%. #User denotes the number of users in the initial training data. Model updates at each chunk with $ch_{size} = 200$ samples, while evaluations are conducted over a sliding window with $w_{size} = w_{step} = 200$ samples.

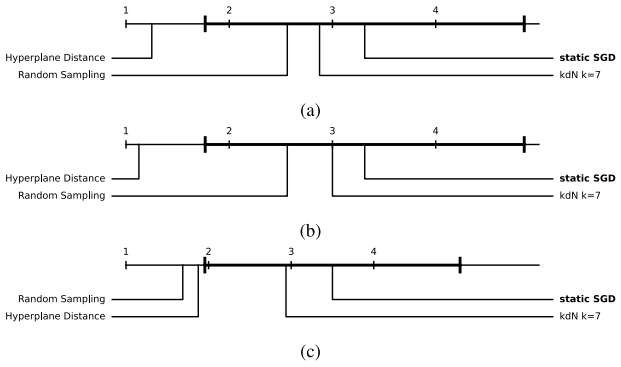


Fig. 7. Critical difference diagrams comparing the stream active learning strategies against the static baseline using the Bonferroni-Dunn post hoc test ($\alpha = 0.05$) on the last-chunk results. Panels correspond to different labeling budgets: (a) 1%, (b) 5%, and (c) 10%. Leftmost methods indicate better performance. All the models with ranks outside the marked interval are significantly different from the static SGD model.

rejected at any labeling budget ($p = 0.289$, $p = 0.256$, and $p = 0.224$ for 1%, 5%, and 10%, respectively). For the latter, the null hypothesis was rejected in all cases ($p = 0.006$, $p = 0.003$, and $p = 0.011$), indicating statistically significant differences among methods at the end of the stream.

Given these results, Bonferroni–Dunn post hoc tests were performed at $\alpha = 0.05$ to compare each AL strategy against the static model using results of the last chunk. Critical difference diagrams [31], shown in Fig. 7, illustrate these comparisons, where methods outside the marked interval differ significantly from the static approach, and leftmost methods indicate lower EER.

Statistical results show that stream HSV achieves performance comparable to static HSV in terms of average EER over the stream, while requiring significantly fewer labeled samples, independently of the active learning strategy and labeling budget employed. By the end of the

stream, methods such as Hyperplane Distance show statistically significant improvements over the baseline, regardless the labeling budget. A similar effect is observed for random sampling at the 10% labeling level. In contrast, although kDN sampling yields better results, the difference is not statistically significant when compared to static SGD.

In summary, experiments demonstrate that the dissimilarity-based stream HSV system progressively improves performance even with minimal initial data and small labeling budgets. Stable results are observed with kDN and random sampling, whereas hyperplane-based sampling shows higher variability. Statistical analysis confirms that significant improvements over the baseline emerge at the end of the stream. Complete tabular results for the last sliding window of incoming signatures, comparing stream HSV to static SGD and SVM classifiers, are available in the supplementary material.

6. Conclusion

This paper presented a dissimilarity-based stream framework for handwritten signature verification that integrates partially labeled data with active learning techniques. The proposed approach demonstrates the feasibility of including active query strategies to enhance verification performance while minimizing annotation costs. By leveraging a dissimilarity-based transformation and writer-independent classification, the system is able to adapt to new signature variations while improving or maintaining performance even with limited labeled data.

Experimental results confirm that random sampling and k-Disagreeing Neighbors (kDN) strategies offer stable and reliable query selection. Our findings highlight the benefits of model-agnostic active learning in biometric applications. Future work may explore semi-supervised learning techniques to further reduce reliance on labeled data and investigate adaptive query strategies that dynamically adjust selection criteria based on stream characteristics and labeling budgets.

CRediT authorship contribution statement

Kecia G. de Moura: Writing – review & editing, Writing – original draft, Software, Investigation, Data curation; **Rafael M.O. Cruz:** Supervision, Methodology, Conceptualization; **Robert Sabourin:** Supervision, Methodology, 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.

Supplementary material

Supplementary material associated with this article can be found in the online version at 10.1016/j.patrec.2026.05.018.

References

[1] 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).
[2] T. Younesian, S. Masoudnia, R. Hosseini, B.N. Araabi, Active transfer learning for Persian offline signature verification, in: 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA), 2019, pp. 234–239.
[3] T.B. Viana, V.L.F. Souza, A.L.I. Oliveira, R.M.O. Cruz, R. Sabourin, A multi-task approach for contrastive learning of handwritten signature feature representations, Expert Syst. Appl. 217 (2023) 119589.
[4] C. Fahy, S. Yang, M. Gongora, Scarcity of labels in non-stationary data streams: a survey, ACM Comput. Surv. 55 (2) (2022). https://doi.org/10.1145/3494832
[5] 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), Springer-Verlag, Berlin, Heidelberg, 2024, 271–286.
[6] V.L.F. Souza, A.L.I. Oliveira, R.M.O. Cruz, R. Sabourin, A white-box analysis on the writer-independent dichotomy transformation applied to offline handwritten signature verification, Expert Syst. Appl. 154 (2020) 113397.
[7] A. Singla, A. Mittal, Exploring offline signature verification techniques: a survey based on methods and future directions, Multimed. Tools Appl. 84 (6) (2025) 2835–2875.
[8] A. Tharwat, W. Schenck, A survey on active learning: state-of-the-art, practical challenges and research directions, Mathematics 11 (4) (2023).
[9] W. Janover, T. Westphal, Signature verification and mail ballots: guaranteeing access while preserving integrity-a case study of California's every vote counts act, Elect. Law J.: Rules Polit. Policy 19 (3) (2020) 321–343.
[10] U. S. Election Assistance Commission, Election administration and voting survey 2022 comprehensive report, 2023.
[11] M.R. Smith, T. Martinez, C. Giraud-Carrier, An instance level analysis of data complexity, Mach. Learn. 95 (2) (2014) 225–256.
[12] I. Deutschmann, P. Nordström, L. Nilsson, Continuous authentication using behavioral biometrics, IT Prof. 15 (4) (2013) 12–15.
[13] S. Mondal, P. Bours, A study on continuous authentication using a combination of keystroke and mouse biometrics, Neurocomputing 230 (2017) 1–22.
[14] L. de Marcos, J.-J. Martínez-Herráiz, J. Junquera-Sánchez, C. Cilleruelo, C. Pages-Arévalo, Comparing machine learning classifiers for continuous authentication on mobile devices by keystroke dynamics, Electronics 10 (14) (2021).
[15] T. Sim, S. Zhang, R. Janakiraman, S. Kumar, Continuous verification using multimodal biometrics, IEEE Trans. Pattern Anal. Mach. Intell. 29 (4) (2007) 687–700.
[16] Z. Qin, P. Zhao, T. Zhuang, F. Deng, Y. Ding, D. Chen, A survey of identity recognition via data fusion and feature learning, Inf. Fusion 91 (2023) 694–712.
[17] A. Morales, D. Morocho, J. Fierrez, R. Vera-Rodriguez, Signature authentication based on human intervention: performance and complementarity with automatic systems, IET Biom. 6 (4) (2017) 307–315.
[18] G. Eskander, E. Granger, Dissimilarity representation for handwritten signature verification, CEUR Workshop Proc. 1022 (2013).
[19] G. Aguiar, B. Krawczyk, A. Cano, A survey on learning from imbalanced data streams: taxonomy, challenges, empirical study, and reproducible experimental framework, Mach. Learn. 113 (7) (2024) 4165–4243.
[20] M. Hameed, R. Ahmad, M.L. Mat Kiah, G. Murtaza, Machine learning-based offline signature verification systems: a systematic review, Signal Process.: Image Commun. 93 (2021) 116139.
[21] V. Losing, B. Hammer, H. Wersing, Incremental on-line learning: a review and comparison of state of the art algorithms, Neurocomputing 275 (2018) 1261–1274.
[22] C. Panagiotakopoulos, P. Tsampouka, The Stochastic gradient descent for the primal L1-SVM optimization revisited, in: Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 65–80.
[23] S. Shalev-Shwartz, Y. Singer, N. Srebro, A. Cotter, Pegasos: primal estimated sub-gradient solver for SVM, Math. Program. 127 (1) (2011) 3–30.
[24] T. Zhai, Y. Gao, H. Wang, L. Cao, Classification of high-dimensional evolving data streams via a resource-efficient online ensemble, Data Min. Knowl. Discov. 31 (5) (2017) 1242–1265.
[25] A.C. Lorena, P.Y.A. Paiva, R.B.C. Prudêncio, Trusting my predictions: on the value of instance-level analysis, ACM Comput. Surv. 56 (7) (2024).
[26] V.L.F. Souza, A.L.I. Oliveira, R.M.O. Cruz, R. Sabourin, Characterization of handwritten signature images in dissimilarity representation space, in: Computational Science – ICCS 2019, Springer International Publishing, Cham, 2019, pp. 192–206.
[27] 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.
[28] 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.
[29] 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.I. Moro-Sancho, MCYT baseline corpus: a bimodal biometric database. IEE proc vis image signal process spec issue biom internet, IEE Proc. - Vis. Image Signal Process. 150 (2003) 395–401.
[30] A. Bifet, R. Gavaldà, G. Holmes, B. Pfahringer, Machine Learning for Data Streams: with Practical Examples in MOA, The MIT Press, 2018.
[31] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1-30.
[32] A. Bifet, G. de Francisci Morales, J. Read, G. Holmes, B. Pfahringer, Efficient online evaluation of big data stream classifiers, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, Association for Computing Machinery, New York, NY, USA, 2015, 59-68.