On Dissimilarity Representation and Transfer Learning for Offline Handwritten Signature Verification

Victor L. F. Souza[1], Adriano L. I. Oliveira[1], Rafael M. O. Cruz[2] and Robert Sabourin[3]
[1]*Centro de Informática - Universidade Federal de Pernambuco*, Recife, Pernambuco, Brazil
Email: vlfs@cin.ufpe.br, alio@cin.ufpe.br
[2]*Stradigi AI*, Montreal, Quebec, Canada
Email: rafaelmenelau@gmail.com
[3]*École de technologie supérieure - Université du Québec*, Montreal, Quebec, Canada
Email: robert.sabourin@etsmtl.ca

***Abstract*—When compared to Writer-Dependent (WD) Handwritten Signature Verification, in which a model is trained for each individual writer, the Writer-Independent (WI) approach offers greater scalability, since only a single model is trained for all users from a dissimilarity space generated by the dichotomy transformation. However, many samples from the dissimilarity space are redundant and have little influence during the training of the verification model. This work investigates whether prototype selection (PS) preprocessing can be used in the space resulting from the dichotomy transformation without degrading the performance of the classifier. Furthermore, an investigation is also performed to examine the use of a WI classifier in a transfer learning scenario, i.e., where the classifier is trained in one dataset, and is used to verify signatures in other datasets. The experiments reported herein show that the use of prototype selection in the dissimilarity space allows a reduction in the complexity of the classifier without degrading its generalization performance. In addition, the results show that the WI classifier is scalable enough to be used in a transfer learning approach, with a resulting performance comparable to that of a classifier trained and tested in the same dataset. An analysis of the results obtained based on the instance hardness (IH) measure and dendrogram diagrams is performed in order to better understand the behavior of the resulting dichotomy transformation.**

***Index Terms*—Offline signature verification, Writer-independent signature verification, Dichotomy transformation, Prototype selection, Instance hardness, Transfer learning.**

I. Introduction

The problem of automatic Handwritten Signature Verification (HSV) can be defined as follows: Given a learning set containing genuine signatures of a set of users, a model is trained to classify the signatures as genuine or forgeries. In offline HSV systems, the signature is acquired as an image after the writing process is completed [1].

Considering the user horizon in HSV problems, if a model is trained for each user, the system is Writer-Dependent (WD). In this case, a binary classifier is trained for each user. Although WD systems achieve good results for the HSV task, requiring a classifier for each user increases the complexity and the cost of the system operations as more users are added [2].

HSV systems used to classify the signatures of any available user in the dataset are known as Writer-Independent (WI) systems. Here, a single model is trained for all users from a dissimilarity space generated by the dichotomy transformation (DT). When compared to the WD approach, WI systems are less complex, but in general provide worse results [1].

Since each dissimilarity vector generated by the DT is formed by the difference between the features of a questioned signature and a reference signature, this approach can increase the number of samples in the WI-HSV scenario. However, many of these samples are redundant, and have little influence for training the verification model. Using prototype selection (PS) techniques in the dissimilarity space may thus allow a reduction of the complexity and the training time of the classifier used without degrading its generalization [3].

In the WI case as well, a single model is trained for all users, and the classification depends solely on the input reference signature. Consequently, a WI-HSV system is more scalable than a writer-dependent one, and could theoretically be used in a transfer learning approach to perform a verification in a dataset other than the one in which it was trained.

Transfer learning methods seek to use sufficient amounts of prior knowledge from other related domains when executing new tasks in the given domain. In general, both domains are under a different distribution [4].

The objective of this paper is to analyze the use of both prototype selection and transfer learning in the context of offline WI-HSV based on dichotomy transformation (DT). In particular, the following points are investigated: (i) What are the main characteristics of the dissimilarity space for WI-HSV? (ii) Given the probable redundancy of samples, can a prototype selection preprocessing be applied without degrading the performance of the classifier? (iii) Is a preprocessing based on a systematic prototype selection technique better than a random subsampling? (iv) Can a WI classifier trained in one dataset be used to verify signatures in the other datasets?

This paper is organized as follows: Section II presents the fundamentals of this work. Section III contains the discussion and the experiments conducted for both the prototype selection and the transfer learning scenarios. In the last section, the conclusion and future works are presented.

II. Fundamentals

A. Handwritten Signature Verification (HSV)

The handwritten signature is one of the oldest accepted biometric characteristic used to verify whether a person is who he/she claims to be. The key task for an offline handwritten signature verification system is deciding whether a given signature image is genuine or a forgery, making it a two-class pattern classification problem [5]. While genuine signatures are those that really belong to the indicated person, forgeries are those created by other people. Forgeries can be broken down into the following types [6]:

- Random forgeries: these happen when the forger neither knows the name of the signer nor the signature pattern.
- Simple forgeries: the forger only has the access to the name of the genuine writer, but does not know the signature pattern.
- Skilled forgeries: the forger knows both the name of the signer and the genuine signature pattern. This results in forgeries that are more similar to genuine signatures.

The current state of the art in feature representation for offline signatures is reported in a paper by Hafemann et al. [5] which uses Deep Convolutional Neural Networks (DCNN) for learning signature representations in a writer-independent manner. The approach based on DCNN tries to learn a new feature space with the most representative properties of the handwritten signatures. As part of a writer-independent approach, the learned representation space is not specific to a single set of users, and is able to use data from as many users as possible.

In the present work, the 2048 features obtained from the FC7 layer of the DCNN, called SigNet, are used as feature vectors [5] (available online[1]).

B. Dichotomy Transformation (DT)

The Dichotomy Transformation (DT), proposed by Cha and Srihari [7], is an approach that allows transforming pattern recognition problems having K-classes, where K is a large value, into a 2-class problem. In this context, the HSV problem can be presented as follows: given a reference signature and a questioned signature, the objective is to determine whether these two signatures were produced by the same writer.

In a more formal definition, let $\mathbf{x}_q$ and $\mathbf{x}_r$ be two feature vectors, respectively from the questioned signature and the reference signature, in the feature space. The distance vector in the dissimilarity space resulting from the Dichotomy Transformation, $\mathbf{u}$, is computed by Equation 1:

$$\mathbf{u}(\mathbf{x}_q, \mathbf{x}_r) = \begin{bmatrix} |x_{q1} - x_{r1}| \\ |x_{q2} - x_{r2}| \\ \vdots \\ |x_{qn} - x_{rn}| \end{bmatrix} \tag{1}$$

where $|\cdot|$ represents the absolute value of the difference, x_{qi} and x_{ri} are the n-th features of the signatures $\mathbf{x}_q$ and $\mathbf{x}_r$, respectively, and n is the number of features [7]. It is worth highlighting that each component of the dissimilarity vector $\mathbf{u}$ is computed from the corresponding components of the vectors $\mathbf{x}_q$ and $\mathbf{x}_r$. Thus, both the distance vector and the feature vectors have the same dimensionality.

As previously noted, in the dissimilarity space, regardless of the number of writers, there are only two classes, namely, (i) the within class w_+ (positive class), composed of distance vectors computed from samples of the same writer (i.e., intraclass distances), and (ii) the between class w_- (negative class), composed of distance vectors computed from samples of different writers (i.e., interclass distances).

Systems based on the DT approach need datasets that have already been transposed into the dissimilarity space to train a dichotomizer (two-class classifier), which will be used to perform the verification task. Generally, the writers that are in the training set are not part of the test set [7].

If each writer presented to the system has more than one reference signature, each comparison between the questioned signature and the reference signatures results in a partial decision, and the final decision is computed based on all partial ones. Intuitively, the greater the number of reference signatures available for this comparison, the more accurate the final decision will be [8].

Formally, when users have more than one reference signature each, the dichotomy transformation is applied between the feature vector $\mathbf{x}_q$ of the questioned signature and the writer's reference set $\{\mathbf{x}_r\}_1^R$, producing a set of dissimilarity vectors $\{\mathbf{u}_r\}_1^R$, where R is the number of signatures in the reference set. For example, if a writer has 3 reference signatures ($R = 3$) and $\{\mathbf{u}_r\}_1^R = \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$, then the dichotomizer evaluates each dissimilarity vector individually and produces a set of partial decisions $\{f(\mathbf{u}_r)\}_1^R$ [6]. The final decision regarding the questioned signature is based on the fusion of all partial decisions by a function $g(\cdot)$. In our case, the output of the classifier is a numerical value, and we use the Max function for fusion [9].

C. Prototype Selection (PS)

Prototype Selection (PS) approaches generally aim to obtain a representative training subset, with a lower number of samples as compared to the original one ($SelectedSubset \subseteq TrainingSet$) [3]. Using the selected subset (from PS) can result in a similar or even higher classification accuracy for new incoming data [3].

According to Pekalska et al. [10], prototype selection is an important aspect that should be considered for dissimilarity-based classification. In their paper, the authors showed that by using a few but well-chosen selected prototypes, it is possible to speed up the classifier training and still achieve a classification performance that is similar to or better than what is obtained by using all the training samples together. The authors also showed that, in general, a systematic prototypes selection approach works better than a random subsampling. To the best of our

[1] http://en.etsmtl.ca/Unites-de-recherche/LIVIA/Recherche-et-innovation/Projets/Signature-Verification

knowledge, this analysis has never been conducted specifically for the dichotomy transformation scenario.

In the present work, the classical Condensed Nearest Neighbors (CNN) approach is used for systematic prototype selection. This approach maintains the instances that are misclassified by a 1-NN classifier, discarding them otherwise [11]. CNN was chosen because its goal is to reduce the dataset size by removing redundant instances, maintaining the samples in the decision boundaries [3].

D. Transfer Learning (TL)

Transfer learning (TL) techniques aim to extract useful information from a source domain and apply it to a target domain. In most cases, there is only one target domain for a transfer learning task, although either single- or multiple-source domains are possible [4].

In general, when dealing with knowledge transfer some problems, such as, data distribution mismatch, may appear. Transfer learning methods (e.g., mining shared patterns from data across different domains) can significantly reduce the difference between the target and the source domains such that the performance of the task in the target domain is improved [12]. To the best of our knowledge, the transfer learning of dichotomy transformation has never been studied.

In their work, Hafemann et al. [5] use the transfer of feature representation in the HSV scenario, as the same networks trained in the GPDS dataset are used for extracting features on MCYT, CEDAR and the BRAZILIAN PUC-PR datasets.

E. Instance Hardness (IH)

The instance hardness (IH) is a metric used both to identify hard to classify instances and to understand why they are hard to classify [13]. Understanding why the instances are misclassified provides an indication of the best preprocessing technique or the best classifier to be used [13]. For instance, in the paper by Cruz et al. [14], the authors use IH to identify the scenarios where ensembles with dynamic selection techniques outperform the K-NN classifier.

In this work, the kDisagreeing Neighbors (kDN) measure is used to estimate the instance hardness. It represents the percentage of instances in an instance's neighborhood that do not share the same label of itself. This metric was chosen because it has the highest correlation with the probability that a given instance is misclassified by different classification methods [13]. The kDN hardness measure is computed by Equation 2:

$$kDN(x_q) = \frac{|x_k : x_k \in KNN(x_q) \wedge label(x_k) \neq label(x_q)|}{K} \tag{2}$$

where $KNN(x_q)$ represents the set of K nearest neighbors of a query instance x_q and x_k represents an instance in its neighborhood. $label(x_q)$ and $label(x_k)$ represent the class labels of the instances x_q and x_k respectively [13].

III. Experiments

The objective of the experiments was to analyze whether: (i) prototype selection preprocessing techniques can be used without degrading the performance of the classifier; (ii) preprocessing based on a systematic prototype selection technique is better than a random selection for the WI-HSV problem; and (iii) WI-SVM trained in the GPDS dataset can be used to verify signatures in the other datasets.

The aim was also to explain these objectives based on the main characteristics of the dissimilarity space resulting from the dichotomy transformation for WI-HSV. The instance hardness distribution of genuine signatures, random forgeries and skilled forgeries, as well as dendrograms, were used to this end.

A. Datasets

The experiments were carried out using the GPDS, BRAZILIAN, MCYT and CEDAR datasets, which are summarized in Table I.

TABLE I
Summary of the datasets used

Dataset Name	Users	Genuine signatures (per user)	Forgeries per user
GPDS Signature 960	881	24	30
Brazilian (PUC-PR)	60+108	40	10 Simple, 10 Skilled
MCYT-75	75	15	15
CEDAR	55	24	24

For the GPDS, we used the GPDS-300 segmentation, and so the Exploitation set ε was composed of the first 300 writers, while the others formed the Development set D.

The Development set segmentation was done considering the methodologies of the papers by Rivard et al. [6] and by Eskander et al [2]. The learning set L was generated using a subset of 14 of the 24 genuine signatures from the development dataset. Hence, the positive class samples were computed using all genuine signatures from every writer, as in Table II. To generate an equivalent number of counterexamples, the *negative class* was formed by using 13 genuine signatures (reference signatures) against 7 random forgeries, with each one selected from a genuine signature of 7 different writers (Table II).

TABLE II
Development set segmentation of the GPDS-300 dataset

Learning set (L)	
Positive Class	Negative Class
Distances between the 14 signatures for each writer (D)	Distances between the 13 signatures for each writer and 7 random signatures from other writers
$581 \cdot 14 \cdot 13/2 = 52,871$ samples	$581 \cdot 13 \cdot 7 = 52,871$ samples

For the BRAZILIAN dataset, the same methodology was used [2], and the division is summarized in Table III. For CEDAR and MCYT datasets, we used a 5x2 fold cross-validation. Hence, as the MCYT dataset has 75 writers, each fold would have 37 or 38 writers. For the training folds, from the 15 genuine signatures of each writer in D, 10 signatures are randomly selected to generate the learning set L (Table

TABLE III
DEVELOPMENT SET SEGMENTATION OF THE BRAZILIAN DATASET

Learning set (L)	
Positive Class	Negative Class
Distances between the 30 signatures for each writer (D)	Distances between the 29 signatures for each writer and 15 random signatures from other writers
$108 \cdot 30 \cdot 29/2 = 46,980$ samples	$108 \cdot 29 \cdot 15 = 46,980$ samples

V). For the CEDAR dataset, the 55 writers were split into 27 or 28 writers per fold. For the training folds, the 24 genuine signatures of each writer in D, 14 signatures are randomly selected to generate the learning set L (Table IV). The other fold is used for testing in both scenarios.

TABLE IV
DEVELOPMENT SET SEGMENTATION OF THE CEDAR DATASET

Learning set (L)	
Positive Class	Negative Class
Distances between the 14 signatures for each writer (D)	Distances between the 13 signatures for each writer and 7 random signatures from other writers
$(27\ or\ 28) \cdot 14 \cdot 13/2$ samples	$(27\ or\ 28) \cdot 13 \cdot 7$ samples

TABLE V
DEVELOPMENT SET SEGMENTATION OF THE MCYT DATASET

Learning set (L)	
Positive Class	Negative Class
Distances between the 10 signatures for each writer (D)	Distances between the 9 signatures for each writer and 5 random signatures from other writers
$(37\ or\ 38) \cdot 10 \cdot 9/2$ samples	$(37\ or\ 38) \cdot 9 \cdot 5$ samples

Considering that each dataset has a different number of writers and signature per writer, and to allow a comparison of the results with the state of the art, the testing set is acquired as in [5].

B. Experimental setup

Before feeding the classifier, the distance vectors **u** (in the dissimilarity space) are standardized by removing the mean and scaling to unit variance. In the transfer learning scenarios, the same normalization from the GPDS is used for the other datasets (so the data is on the same scale).

The Support Vector Machine (SVM) is considered as one of the best classification methods for both WD and WI signature verification tasks [1]. In this paper, the SVM is used as a writer-independent classifier with the following settings: RBF kernel, $\gamma = 2^{-11}$ and $C = 1.0$ [9]. The predicted confidence scores for samples are used as classifier outputs. The confidence score for a sample is the signed distance from that sample to the classifier's hyperplane [5].

All data were randomly selected, and a different SVM was trained for each replication (ten replications were performed for each experimental configuration). Considering the Prototype Selection techniques, for the Condensed Nearest Neighbors, K_{CNN} was set to 1, as in the original algorithm [11]. For the instance hardness analysis, the neighborhood size $K = 7$ was used for the estimation of the kDN [14].

The performance evaluation of the classification methods is based on the Equal Error Rate (EER) metric, using user thresholds (considering just the genuine signatures and the skilled forgeries) [5]. In the paper by Souza et al. [9], for the tested dataset, the best results are generally obtained using the highest number of references and Max as the fusion function. Therefore, only this approach is considered in the work. To evaluate the effectiveness of the results, we conducted the Wilcoxon paired signed-rank test with a 5% level of significance to confirm whether the two methods were significantly different.

C. Using Prototype Selection

The following experiments evaluate the application of prototype selection before training the SVM. The $\%\_SVM$ represents the models with uniform random subsampling of the training set. We use 1.0%, 5.0% and 10.0% of the original training set. The Condensed Nearest Neighbors is called CNN_SVM in the tables.

1) GPDS-300 dataset: Table VI presents a comparative analysis of the results obtained by the SVMs (with and without prototype selection) versus those obtained with state of the art models, considering the EER metric. Tables VII and VIII respectively present the comparative analysis of the number of samples and the number of support vectors (SV) obtained by the SVMs (with and without prototype selection), for the GPDS-300 dataset.

TABLE VI
COMPARISON OF EER WITH THE STATE OF THE ART IN THE GPDS-300 DATASET, USING MAX FUNCTION (ERRORS IN %)

Type	Model	#references	EER
WD	Soleimani et al. [15]	10	20.94
WD	Hafemann, Sabourin and Oliveira [16]	12	12.83
WD	Hafemann et al. [5]	5	3.92 (0.18)
WD	Hafemann et al. [5]	12	3.15 (0.18)
WI	SVM_{max}	12	3.69 (0.18)
WI	$1\%\_SVM_{max}$	12	3.54 (0.26)
WI	$5\%\_SVM_{max}$	12	3.62 (0.32)
WI	$10\%\_SVM_{max}$	12	3.48 (0.12)
WI	CNN_SVM_{max}	12	3.47 (0.15

TABLE VII
COMPARISON OF THE NUMBER OF TRAINING SAMPLES IN THE GPDS-300 DATASET

Model	#Positive Samples	#Negative Samples	#Retained Samples (%)
SVM	52871	52871	100.00 (0.00)
$1\%\_SVM$	531.70 (17.04)	526.30 (17.04)	1.00 (0.00)
$5\%\_SVM$	2648.10 (24.78)	2639.90 (24.78)	5.00 (0.00)
$10\%\_SVM$	5289.30 (31.69)	5285.70 (31.69)	10.00 (0.00)
CNN_SVM	345.90 (15.25)	4437.80 (125.11)	4.52 (0.13)

As presented in Tables VI and VII, the use of the prototype selection methods allows the SVM to be trained with a much smaller number of samples, and to still provide comparable results. This also results in a large reduction in the number in the support vectors used by the SVM (Table VIII), which in turn reduces the complexity and computational cost of training a SVM in the offline WI-HSV context.

In Table VI, a simple random subsampling with 1% of the training samples provides similar results to what is obtained

TABLE VIII
COMPARISON OF THE NUMBER OF SUPPORT VECTORS (SV) IN THE GPDS-300 DATASET

Model	#SV	#Positive SV	#Negative SV
SVM	3398.40 (95.01)	1640.30 (45.90)	1758.10 (53.46)
$1\%\_SVM$	194.60 (8.92)	78.70 (4.61)	115.90 (9.87)
$5\%\_SVM$	481.30 (16.78)	208.50 (12.15)	272.80 (11.39)
$10\%\_SVM$	720.90 (23.64)	309.80 (9.35)	411.10 (16.88)
CNN_SVM	928.20 (28.44)	312.90 (12.32)	615.30 (19.34)

with the SVM trained with the complete training set. This shows that the samples resulting from the dichotomy transformation are redundant for this dataset.

For Table VI, considering the WD model from Hafemann et al. [5] for the GPDS-300 dataset, both the models, with and without preprocessing, obtained comparable results for the EER metric, even operating in a writer-independent fashion. When compared to the other models, the proposed approach obtained better results.

Given these results, we can see that for the GPDS-300 dataset, the dichotomy transformation was able to increase the number of samples in the WI-HSV scenario, and yet many of them were redundant. This therefore means that the use of prototype selection in the dissimilarity space allowed a reduction of the complexity of the classifier used without degrading its results.

2) BRAZILIAN dataset: Tables IX, X and XI respectively present, a comparative analysis of the classification metrics, the number of samples and the number of support vectors (SV) obtained by the SVMs (with and without prototype selection) in the BRAZILIAN dataset.

TABLE IX
COMPARISON OF EER WITH THE STATE OF THE ART IN THE BRAZILIAN DATASET, USING MAX FUNCTION (ERRORS IN %)

Type	Model	#references	EER
WD	Hafemann, Sabourin and Oliveira [16]	15	4.17
WD	Hafemann et al. [5]	5	2.92 (0.44)
WD	Hafemann et al. [5]	15	2.07 (0.63)
WD	Hafemann et al. [5]	30	2.01 (0.43)
WI	SVM_{max}	30	1.47 (0.36)
WI	$1\%\_SVM_{max}$	30	1.21 (0.45)
WI	$5\%\_SVM_{max}$	30	1.19 (0.42)
WI	$10\%\_SVM_{max}$	30	1.23 (0.51)
WI	CNN_SVM_{max}	30	1.26 (0.33)

Much as in the GPDS-300 scenario, Tables IX and X show that a simple random subsampling with 1% of the samples maintains similar results as those obtained with the SVM trained with the complete training set.

Once again, this demonstrates that the samples resulting from the dichotomy transformation are redundant for this database. The data from Brazilian dataset are probably more redundant when compared to what we have in the GPDS-300 dataset. This can be observed from the greater reduction secured by the CNN approach: while 4.52% of the samples are needed to represent the border region in the GPDS-300, only 1.47% is needed for the BRAZILIAN dataset.

Still in Table IX, for this dataset, even operating in a writer-independent fashion, both the models with and without preprocessing obtained better results considering the EER metric, when compared to the other WD models.

TABLE X
COMPARISON OF THE NUMBER OF TRAINING SAMPLES IN THE BRAZILIAN DATASET

Model	#Positive Samples	#Negative Samples	#Retained Samples (%)
SVM	46980	46980	100.00 (0.00)
$1\%\_random$	474.20 (14.60)	465.80 (14.60)	1.00 (0.00)
$5\%\_random$	2336.50 (35.63)	2361.50 (35.63)	5.00 (0.00)
$10\%\_random$	4681.60 (38.21)	4714.40 (38.21)	10.00 (0.00)
CNN_SVM	379.30 (41.22)	1005.10 (33.65)	1.47 (0.07)

TABLE XI
COMPARISON OF THE NUMBER OF SUPPORT VECTORS (SV) IN THE BRAZILIAN DATASET

Model	#SV	#Positive SV	#Negative SV
SVM	3368.80 (72.96)	1627.40 (40.70)	1741.40 (41.29)
$1\%\_SVM$	259.70 (17.25)	92.70 (7.44)	167.00 (11.09)
$5\%\_SVM$	688.00 (33.79)	267.80 (13.67)	420.20 (28.61)
$10\%\_SVM$	1014.20 (43.57)	420.20 (14.23)	594.00 (31.38)
CNN_SVM	658.40 (34.84)	261.00 (18.77)	397.40 (20.93)

3) MCYT dataset: Tables XII, XIII and XIV respectively present a comparative analysis on the classification metrics, the number of samples and the number of support vectors (SV) obtained by the SVMs (with and without prototype selection) in the MCYT dataset.

TABLE XII
COMPARISON OF EER WITH THE STATE OF THE ART IN THE MCYT DATASET, USING MAX FUNCTION (ERRORS IN %)

Type	Model	#references	EER
WD	Gilperez et al. [17]	10	6.44
WD	Wen et al. [18]	5	15.02
WD	Vargas et al. [19]	10	7.08
WD	Ooi et al. [20]	10	9.87
WD	Soleimani et al [15]	10	9.86
WD	Hafemann et al. [5]	10	2.87 (0.42)
WI	SVM_{max}	10	2.73 (0.20)
WI	$1\%\_SVM_{max}$	10	3.67 (0.11)
WI	$5\%\_SVM_{max}$	10	3.27 (0.26)
WI	$10\%\_SVM_{max}$	10	3.19 (0.20)
WI	CNN_SVM_{max}	10	2.99 (0.16)

As presented in Table XII, unlike with the CNN, the use of random subsampling resulted in the degradation of the performance of the classifier. Used as the prototype selection method, the Condensed Nearest Neighbors provided results comparable to those obtained with the SVM trained with all the data; additionally the CNN allowed the SVM to be trained with only about 8% of the training samples (as presented in Table XIII). This also resulted in an almost 28% reduction in the number of the support vectors used by the SVM (Table XIV). Unlike with random subsampling, using the Condensed Nearest Neighbors allowed more attention to be paid to border samples, which removed the need to store more instances than were necessary for an accurate generalization.

Also in Table XII, for the MCYT dataset, when compared to the other models, the proposed approach obtained better results for the EER metric. The only exception was for the comparison with the model proposed in Hafemann et al. [5].

TABLE XIII
COMPARISON OF THE NUMBER OF TRAINING SAMPLES IN THE MCYT DATASET

Model	#Positive Samples	#Negative samples	#Retained Samples (%)
SVM	1687.50 (22.50)	1687.50 (22.50)	100.00 (0.00)
$1\%\_random$	16.78 (3.29)	17.72 (3.13)	1.00 (0.00)
$5\%\_random$	83.22 (6.19)	85.78 (6.00)	5.00 (0.00)
$10\%\_random$	167.78 (9.02)	169.72 (9.51)	10.00 (0.00)
CNN_SVM	38.29 (5.19)	224.93 (21.08)	7.80 (0.75)

TABLE XIV
COMPARISON OF THE NUMBER OF SUPPORT VECTORS (SV) IN THE MCYT DATASET

Model	#SV	#Positive SV	#Negative SV
SVM	567.77 (38.61)	251.29 (18.93)	316.48 (22.14)
$1\%\_SVM$	32.47 (1.66)	14.78 (2.32)	17.69 (3.09)
$5\%\_SVM$	105.63 (5.61)	42.54 (3.57)	63.09 (4.56)
$10\%\_SVM$	161.89 (9.79)	64.27 (4.42)	97.62 (7.47)
CNN_SVM	160.49 (15.66)	38.22 (5.14)	122.27 (12.53)

4) CEDAR dataset: Tables XV, XVI and XVII respectively present a comparative analysis on the classification metrics, the number of samples and the number of support vectors (SV) obtained by the SVMs (with and without prototype selection) in the CEDAR dataset.

TABLE XV
COMPARISON OF EER WITH THE STATE OF THE ART IN THE CEDAR DATASET, USING MAX FUNCTION (ERRORS IN %)

Type	Model	#references	EER
WI	Kumar et al. [21]	1	11.81
WI	Kumar et al. [22]	1	8.33
WD	Hafemann et al. [5]	12	4.76 (0.36)
WI	SVM_{max}	12	5.78 (0.38)
WI	$1\%\_SVM_{max}$	12	7.22 (0.27)
WI	$5\%\_SVM_{max}$	12	6.45 (0.23)
WI	$10\%\_SVM_{max}$	12	6.02 (0.32)
WI	CNN_SVM_{max}	12	5.86 (0.50)

In Table XV, for the CEDAR dataset while the use of random subsampling resulted in the degradation of the model, using the CNN did not affect the performance of the WI classifier. Used as the prototype selection method, the Condensed Nearest Neighbors provided results comparable to those obtained with the SVM trained with all the data; additionally the CNN allowed the SVM to be trained with only about 3% of the training samples (Table XVI). This also results in an almost 18% reduction in the number of the support vectors used by the SVM (Table XVII).

TABLE XVI
COMPARISON OF THE NUMBER OF TRAINING SAMPLES IN THE CEDAR DATASET

Model	#Positive Samples	#Negative Samples	#Retained Samples (%)
SVM	2502.50 (45.50)	2502.50 (45.50)	100.00 (0.00)
$1\%\_random$	24.81 (3.48)	25.69 (3.50)	1.00 (0.00)
$5\%\_random$	124.31 (7.85)	126.19 (7.67)	5.00 (0.00)
$10\%\_random$	251.07 (12.57)	249.93 (11.72)	910.00 (0.00)
CNN_SVM	30.78 (7.43)	115.13 (18.26)	2.91 (0.49)

TABLE XVII
COMPARISON OF THE NUMBER OF SUPPORT VECTORS (SV) IN THE CEDAR DATASET

Model	#SV	#Positive SV	#Negative SV
SVM	676.37 (64.57)	390.30 (35.63)	286.07 (32.22)
$1\%\_SVM$	39.46 (2.97)	14.45 (2.03)	25.01 (2.98)
$5\%\_SVM$	117.60 (10.30)	40.63 (4.33)	76.97 (7.32)
$10\%\_SVM$	181.44 (13.98)	65.09 (5.68)	116.35 (10.16)
CNN_SVM	119.75 (19.37)	30.66 (7.29)	89.09 (13.57)

Still in Table XV, for this dataset, the proposed approach obtained worse results when compared to the model proposed by Hafemann et al. [5] and better results in the comparison with the others WI classifiers. However, the comparative results were obtained by a model using just one reference signature.

Given the above results for the tested datasets, the dichotomy transformation was thus able to increase the number of samples in the offline WI-HSV scenario; however, many of these samples are redundant. Using prototype selection in the dissimilarity space allowed a reduction of the complexity of the classifier used without degrading its performance. Furthermore, using a systematic PS, such as the CNN, allows more attention to be paid to border samples. Consequently, prototype selection may thus be used without degrading the performance of the WI classifier, while removing the need to store more instances than are necessary for an accurate generalization.

Unlike with the CEDAR dataset, the models with and without preprocessing for the other datasets obtained results comparable to those of the WD models for the EER metric, even operating in a writer-independent fashion.

D. Using Transfer Learning and Prototype Selection

For the BRAZILIAN, MCYT and CEDAR datasets, in addition to investigating the use of prototype selection, we also analyze if a WI-SVM trained in the GPDS can be used to verify signatures from other datasets, akin to a transfer learning [12]. The associated results are presented in Table XVIII.

In our scenario, no adaptation is required in the classifier or in the features. We only get the WI-SVM trained in the GPDS and use it to verify signatures in the other datasets ($GPDS_{max}$ results in Table XVIII). SVM_{max} results in this table were obtained by training and testing the classifier on the same dataset. It should be recalled that: (i) all datasets have the same number of features; (ii) the features used for all datasets are based on the Convolutional Neural Network trained in the GPDS, and (iii) the same normalization/standardization is used, and therefore, all data are within the same interval.

In Table XVIII, for the BRAZILIAN and MCYT datasets, the WI-SVM trained in the GPDS-300 obtained results comparable to the SVMs being trained and tested on their own dataset, for the EER metric. More interesting results are presented for the CEDAR dataset, since the WI-SVM was trained in another dataset, and still obtained better results than the classifiers trained and tested on the same dataset. These results also show that using the CNN for transfer learning (CNN_GPDS_{max}) slightly improved the results versus the case with transfer learning without PS ($GPDS_{max}$).

We will now look at why transfer learning works without the need for any additional transfer method based on a dendrogram

TABLE XVIII

COMPARISON OF EER FOR THE MODELS WITH AND WITHOUT TRANSFER LEARNING ON THE BRAZILIAN, MCYT AND CEDAR DATASETS, USING MAX FUNCTION (ERRORS IN %)

Dataset	Model	#references	EER
BRAZILIAN	SVM_{max}	30	1.47 (0.36)
	CNN_SVM_{max}	30	1.26 (0.33)
	$GPDS_{max}$	30	1.35 (0.40)
	CNN_GPDS_{max}	30	1.11 (0.37)
MCYT	SVM_{max}	10	2.73 (0.20)
	CNN_SVM_{max}	10	2.99 (0.16)
	$GPDS_{max}$	10	2.97 (0.20)
	CNN_GPDS_{max}	10	2.89 (0.13)
CEDAR	SVM_{max}	12	5.78 (0.38)
	CNN_SVM_{max}	12	5.86 (0.50)
	$GPDS_{max}$	12	3.42 (0.28)
	CNN_GPDS_{max}	12	3.32 (0.22)

(a hierarchical tree diagram that provides a visualization of the distances between clusters and sub-clusters [23]).

To allow an adequate visualization, the following methodology was performed: 2 samples from each signature type in the Condensed Nearest Neighbors dissimilarity space were randomly selected and the dendrogram was plotted pairwise, comparing the GPDS with the other datasets (BRAZILIAN, MCYT and CEDAR). As the samples were obtained after CNN preprocessing, only the border samples were considered:

- Samples with index 0 and 1: represent the genuine signatures from the GPDS dataset.
- Samples with index 2 and 3: represent the genuine signatures from the comparative dataset.
- Samples with index 4 and 5: represent the forgeries from the GPDS dataset.
- Samples with index 6 and 7: represent the forgeries from the comparative dataset.

Figure 1 shows the dendrogram for the BRAZILIAN dataset (the dendrograms for MCYT and CEDAR have a similar behavior), with the x-axis representing the samples, and the y-axis, the distances obtained.

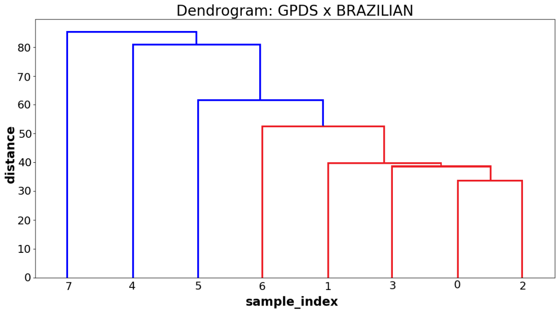

Fig. 1. Dendrogram considering the GPDS and the BRAZILIAN datasets

As shown in Figure 1, the genuine signatures from the BRAZILIAN dataset (indexes 2 and 3) are close to the genuine signatures of the GPDS (indexes 0 and 1). They are therefore close to each other in the dissimilarity space and than the decision frontier for BRAZILIAN datasets should be close to the ones used in the GPDS dataset.

This result in the dissimilarity space generated by the dichotomy transformation explains why the WI-SVM trained in the GPDS can be used to verify signatures in the other datasets without any further transfer adaptation in the offline WI-HSV context.

E. Instance hardness analysis

In this section, we are going to analyze the results obtained by using the instance hardness measure.

Hafemann et al. [5] performed an analysis to examine the local structure of the learned feature space (WD), using the t-SNE algorithm in a subset of the development set of the GPDS-300 dataset, called the validation set for verification V_v. Figure 2 herein is the same as Fig. 5 (b) in their paper [5], and is going to be used here to describe our feature space (as we are using the same features).

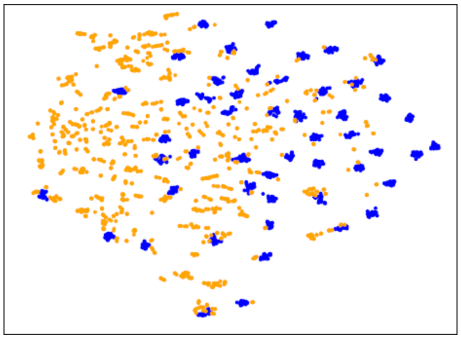

Fig. 2. t-SNE 2D projections of the feature vectors from the 50 users in the validation set for verification V_v. The blue points represent genuine signatures and the orange ones represent skilled forgeries

As can be seen in Figure 2, (i) genuine signatures from different users are clustered and occupy different regions of the feature space; (ii) for some writers, the model achieves a good separation between skilled forgeries and genuine signatures, but this is not the case for all writers, and (iii) some writers still have skilled forgeries which are close to genuine signatures.

With regard to the dissimilarity space representation: (i) signatures that are close in the feature space will be close to the origin in the dissimilarity space, and (ii) the further away two signatures are in the feature space, the farther the vector resulting from the dichotomy transformation will be from the origin [7]. Based on the feature space shown in Figure 2, it is expected that the resulting dissimilarity space will have the following characteristics:

- C_1: Since genuine signatures from the writers form dense clusters in the feature space, positive samples will be close to the origin, forming a dense cluster in the dissimilarity space.
- C_2: As random forgeries are genuine signatures from other writers and different writers occupy different regions of the feature space, negative samples from random forgeries will be far from the origin of the dissimilarity space.
- C_3: For the writers with a larger separation between skilled forgeries and genuine signatures, negative samples will be far from the origin in the dissimilarity space.

- C_4: For the writers that have skilled forgeries close to the genuine signatures, negative samples will be closer to the origin in the dissimilarity space (when compared to the other negative samples), and may even be within the space occupied by the positive samples.

To show that this behavior is actually present, we analyze the instance hardness of the samples in the dissimilarity space using the kDN metric (Eq. 2) in the validation set for verification V_v. A similar methodology as the one applied to obtain the exploitation dataset (section III-A) is used here to obtain the dissimilarity space: (i) the reference set R is composed of just 1 (one) randomly selected genuine signature from each writer of the V_v set, and (ii) the questioned set Q is composed of 10 of the remaining genuine signatures and 10 skilled forgeries from each writer, plus 10 random forgeries, each one selected from a genuine signature of 10 different writers.

Figures 3, 4 and 5 present, for the GPDS dataset, the histograms of the instance hardness considering: (i) all the data, (ii) just positive samples and negative samples from random forgeries, and (iii) just positive samples and negative samples from skilled forgeries, respectively.

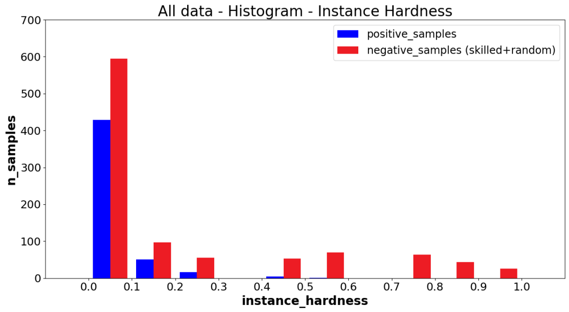

Fig. 3. Instance hardness considering all selected data from the V_v segmentation of GPDS dataset

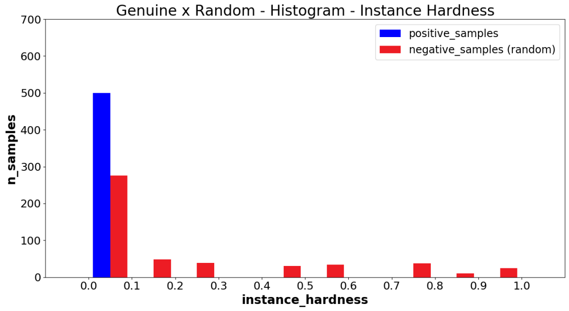

Fig. 4. Instance hardness considering only the positive samples and negative samples (random forgeries) from the V_v segmentation of GPDS dataset

As can be seen in Figure 3, for almost all the positive samples, $IH < 0.3$. So, in the dissimilarity space, since we are considering the kDN with $K = 7$, at least 5 of the 7 neighbors of the positive samples are from the positive class itself (C_1).

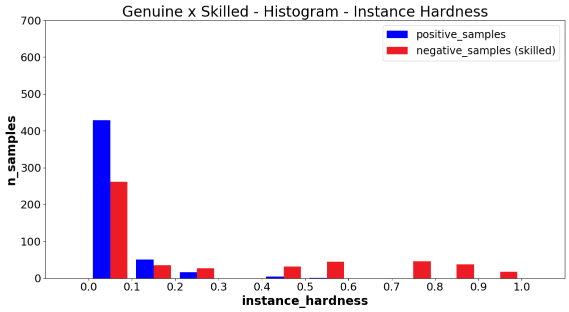

Fig. 5. Instance hardness considering only the positive samples and negative samples (skilled forgeries) from the V_v segmentation of GPDS dataset

As shown in Figure 4, the following points can be seen when considering just the positive samples and the negative samples from the random forgeries: (i) The IH of all the positive samples are in the IH = 0.0 bin. Hence, for this scenario, all the neighbors of the positive samples are from the positive class itself. For this to occur, the positive samples should be concentrated in a dense region of the dissimilarity space, and no positive samples can go to the negative side of the space. Additionally, no negative sample is within the positive cluster (C_1). (ii) The IH of the negative samples are arranged along the histogram, and so the negative samples (random) should be in a sparse region of the dissimilarity space, and some samples should be in a region closer to the dense positive region of the space, since some samples have IH = 1.0 (C_2).

It is worth noting that the representation was able to actually separate positive samples from negative (random) ones, as all the positive samples were in the IH = 0.0 bin, i.e., there was no class overlap.

As can be seen in Figure 5, all the positive samples with $IH \neq 0.0$ from Figure 3 are derived from skilled forgeries. Thus, here, unlike in the negative samples (random) scenario, there should be class overlapping in the dissimilarity space (C_4). This behavior is expected, since, in theory, the skilled forgeries are more similar to the genuine ones, when compared to random forgeries.

The following aspects must also be highlighted: (i) as the positive samples are concentrated on the left side and the negative samples (skilled) are arranged along the histogram, the negative samples should be more sparse than the positive ones in the dissimilarity space (C_3), and (ii) as the negative samples have samples with higher IH, the overlap of the classes should be in the positive region of the dissimilarity space (C_4).

If this same methodology is applied to the rest of the Development dataset (i.e., for the other 531 writers), the data will have a similar IH behavior with a larger number of samples. Making a uniform random selection to pick up the same number of samples as in V_v and performing the Kolmogorov-Smirnov test with a 5% level of significance, we see that both scenarios are drawn from the same continuous distribution in all scenarios. Therefore, the validation set for verification is representative of the Development set.

Generally speaking, positive samples are located in a dense cluster close to the origin and the negative samples are scattered throughout the dissimilarity space. Moreover, the clusters are disjointed, with a small overlap area, based on the concentration of the IH with low values. Considering that hard to classify samples are in the border region, the use of a condensation PS technique such as CNN has been shown to produce good experimental results because it retains samples in the decision boundaries [3]. This IH analysis is also in line with the findings from the previous section regarding the use of transfer learning.

IV. Conclusion

In this work, we evaluated the use of prototype selection and of a WI-SVM in a transfer learning approach applied to the space resulting from dichotomy transformation. The dendrogram approach was used to analyze the results of transfer learning. Moreover, we conducted an analysis using an instance hardness measure in order to better understand the behavior of the resulting space.

The experimental results showed that, in the transfer learning scenario, with the features used, a WI-SVM trained in the GPDS can be employed to verify signatures in the other datasets without any further transfer adaptation in the WI-HSV context and still obtain similar results when compared to both WD and WI classifiers trained and tested in their own datasets.

Additionally, dichotomy transformation is able to increase the number of samples in the offline WI-HSV scenario, but many of the samples become redundant. By using prototype selection, it is therefore possible to speed up the classifier training and still achieve a classification performance that is similar to or better than what is obtained by using all the training samples. Even being a classic and simple technique, the Condensed Nearest Neighbors [11] applied systematically was able to select fewer prototypes and still maintain high performance levels when compared to both the SVM trained with the complete original training set and the random subsampling approach.

Analyses performed using the IH measure and the dendrogram have shown that, in general, positive samples are located in a dense cluster close to the origin, and negative ones are scattered throughout the dissimilarity space generated by the dichotomy transformation.

Future works may include: (i) a study of feature selection in the dissimilarity space, (ii) an adaptation of the WI classifier over time, and (iii) the calculation of writer-dependent decision thresholds.

Acknowledgment

This work was supported by the FACEPE (Fundação de Amparo à Ciência e Tecnologia de Pernambuco), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and the École de Technologie Supérieure (ÉTS Montréal).

References

[1] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Offline handwritten signature verification—literature review," in *Image Processing Theory, Tools and Applications (IPTA), 2017 Seventh International Conference on*. IEEE, 2017, pp. 1–8.

[2] G. S. Eskander, R. Sabourin, and E. Granger, "Hybrid writer-independent–writer-dependent offline signature verification system," *IET biometrics*, vol. 2, no. 4, pp. 169–181, 2013.

[3] S. Garcia, J. Derrac, J. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE T PATTERN ANAL*, vol. 34, no. 3, pp. 417–435, 2012.

[4] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE T NEUR NET LEAR*, vol. 26, no. 5, pp. 1019–1034, 2015.

[5] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," *Pattern Recognition*, vol. 70, pp. 163–176, 2017.

[6] D. Rivard, E. Granger, and R. Sabourin, "Multi-feature extraction and selection in writer-independent off-line signature verification," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 16, no. 1, pp. 83–103, 2013.

[7] S.-H. Cha and S. N. Srihari, "Writer identification: statistical analysis and dichotomizer," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2000, pp. 123–132.

[8] D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, "Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers," *Pattern Recognition*, vol. 43, no. 1, pp. 387–396, 2010.

[9] V. L. Souza, A. L. Oliveira, and R. Sabourin, "A writer-independent approach for offline signature verification using deep convolutional neural networks features," *arXiv preprint arXiv:1807.10755*, 2018.

[10] E. Pekalska, R. P. Duin, and P. Paclik, "Prototype selection for dissimilarity-based classifiers," *Pattern Recognition*, vol. 39, no. 2, pp. 189–208, 2006.

[11] P. Hart, "The condensed nearest neighbor rule (corresp.)," *IEEE transactions on information theory*, vol. 14, no. 3, pp. 515–516, 1968.

[12] S. J. Pan, Q. Yang *et al.*, "A survey on transfer learning," *IEEE T KNOWL DATA EN*, vol. 22, no. 10, pp. 1345–1359, 2010.

[13] M. R. Smith, T. Martinez, and C. Giraud-Carrier, "An instance level analysis of data complexity," *Machine learning*, vol. 95, no. 2, pp. 225–256, 2014.

[14] R. M. Cruz, H. H. Zakane, R. Sabourin, and G. D. Cavalcanti, "Dynamic ensemble selection vs k-nn: why and when dynamic selection obtains higher classification performance?" in *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2017, pp. 1–6.

[15] A. Soleimani, B. N. Araabi, and K. Fouladi, "Deep multitask metric learning for offline signature verification," *Pattern Recognition Letters*, vol. 80, pp. 84–90, 2016.

[16] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Writer-independent feature learning for offline signature verification using deep convolutional neural networks," in *IEEE IJCNN (2016)*. IEEE, 2016, pp. 2576–2583.

[17] A. Gilperez, F. Alonso-Fernandez, S. Pecharroman, J. Fierrez, and J. Ortega-Garcia, "Off-line signature verification using contour features," in *11th International Conference on Frontiers in Handwriting Recognition, Montreal, Quebec-Canada, August 19-21, 2008*. CENPARMI, Concordia University, 2008.

[18] J. Wen, B. Fang, Y. Y. Tang, and T. Zhang, "Model-based signature verification with rotation invariant features," *Pattern Recognition*, vol. 42, no. 7, pp. 1458–1466, 2009.

[19] J. F. Vargas, M. A. Ferrer, C. Travieso, and J. B. Alonso, "Off-line signature verification based on grey level information using texture features," *Pattern Recognition*, vol. 44, no. 2, pp. 375–385, 2011.

[20] S. Y. Ooi, A. B. J. Teoh, Y. H. Pang, and B. Y. Hiew, "Image-based handwritten signature verification using hybrid methods of discrete radon transform, principal component analysis and probabilistic neural network," *Applied Soft Computing*, vol. 40, pp. 274–282, 2016.

[21] R. Kumar, L. Kundu, B. Chanda, and J. Sharma, "A writer-independent off-line signature verification system based on signature morphology," in *Proc. of the 1st Int. Conf. on Intelligent Interactive Technologies and Multimedia*. ACM, 2010, pp. 261–265.

[22] R. Kumar, J. Sharma, and B. Chanda, "Writer-independent off-line signature verification using surroundedness feature," *Pattern recognition letters*, vol. 33, no. 3, pp. 301–308, 2012.

[23] F. Janssens, L. Zhang, B. De Moor, and W. Glänzel, "Hybrid clustering for validation and improvement of subject-classification schemes," *Information Processing & Management*, vol. 45, no. 6, pp. 683–702, 2009.