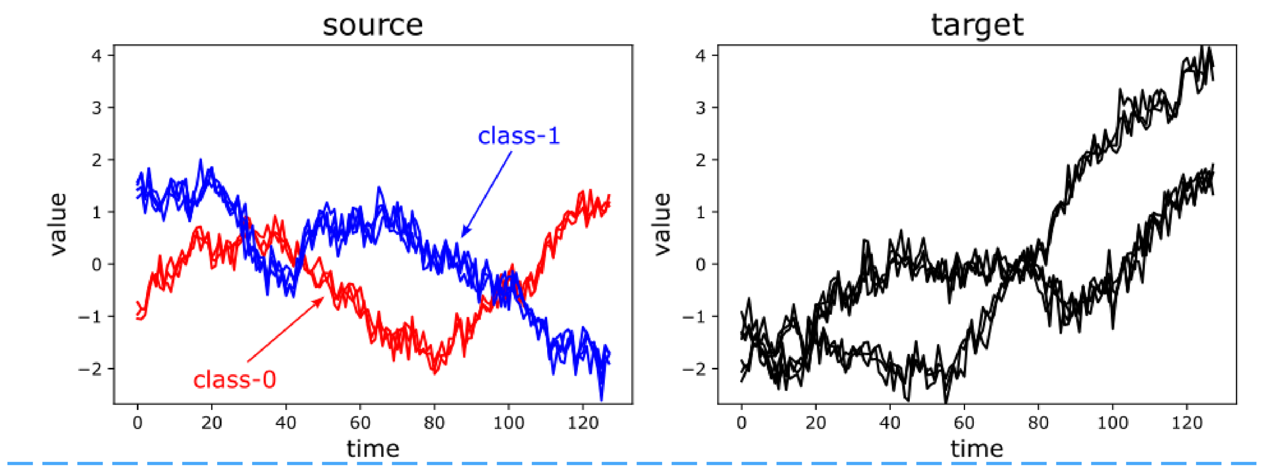
A benchmark on Deep Unsupervised Domain Adaptation (UDA) for Time Series Classification

Introduction & Context

- Unsupervised Domain Adaptation (UDA) aims to harness labeled source data to train models for unlabeled target data.
- A comprehensive benchmark is done to evaluate UDA techniques for times series classification, with a focus on deep learning methods.



UDA approaches considered

Deep UDA algorithms rely on two main components : a backbone encoding the input into a domain-invariant latent space and a classifier.

1. **Baseline** : InceptionTime is used as a baseline with no adaptation to compare with other UDA approaches. Its backbone Inception is used for other UDA algorithms.
2. **Adversarial domain adaptation** : This technique uses a discriminator for distinguishing between source and target samples, while the backbone learns simultaneously how to fool the discriminator (VRADA & CoDats).
3. **Contrastive learning :** This technique aims at aligning the predictions made by the model for pairs of samples coming from two different domains (CoTMix).
4. **Frequency domain analysis**: This technique analyzes temporal features and frequency features of the time series separately (Raincoat).

$\mathcal{L}_C$: classification loss
$\mathcal{L}_A$: adversarial loss
$\mathcal{L}_{VRNN}$: VRNN loss
$\mathcal{L}_R$: reconstruction loss
$\mathcal{L}_{Contrastive}$: contrastive
H: entropy
$\mathcal{L}_{Sinkhorn}$ Sinkhorn divergence

Algorithms	Backbone	Other Modules	Loss function
VRADA	VRNN	Discriminator	$\mathcal{L}_C + \mathcal{L}_A + \mathcal{L}_{\text{VRNN}}$
CoDATS	1D CNN	Discriminator	$\mathcal{L}_C + \mathcal{L}_A$
InceptionDANN	Inception	Discriminator	$\mathcal{L}_C + \mathcal{L}_A$
InceptionCDAN	Inception	Discriminator, Multilinear Map	$\mathcal{L}_C + \mathcal{L}_A$
CoTMix	1D CNN	Temporal Mixup	$\mathcal{L}_C + \mathcal{L}_{\text{Contrastive}} + H$
InceptionMix	Inception	Temporal Mixup	$\mathcal{L}_C + \mathcal{L}_{\text{Contrastive}} + H$
Raincoat	1D CNN	Frequency encoder, Decoder	$\mathcal{L}_C + \mathcal{L}_R + \mathcal{L}_{\text{Sinkhorn}}$
InceptionRain	Inception	Frequency encoder, Decoder	$\mathcal{L}_C + \mathcal{L}_R + \mathcal{L}_{\text{Sinkhorn}}$
InceptionTime	Inception	-	$\mathcal{L}_C$
OTDA	-	Transport map	-

Hyperparameter Tuning

The aim of hyperparameter tuning is to find the hyperparameters that lead to a model minimizing the target risk. This task is particularly difficult in UDA due to the absence of labels in target. Three approaches are considered.

1. **Target risk:** It relies on the empirical target risk to select models, computed on target labels.

$$\hat{\mathcal{R}}_T(f) = \frac{1}{n_T}\sum_{i=1}^{n_T}\mathcal{L}(y_i^T, f(X_i^T))$$

2. **Source risk :** It relies on the empirical source risk.

$$\hat{\mathcal{R}}_S(f) = \frac{1}{n_S}\sum_{i=1}^{n_S}\mathcal{L}(y_i^S, f(X_i^S))$$

3. Importance Weighted Cross Validation (IWCV) : It estimates the target risk.

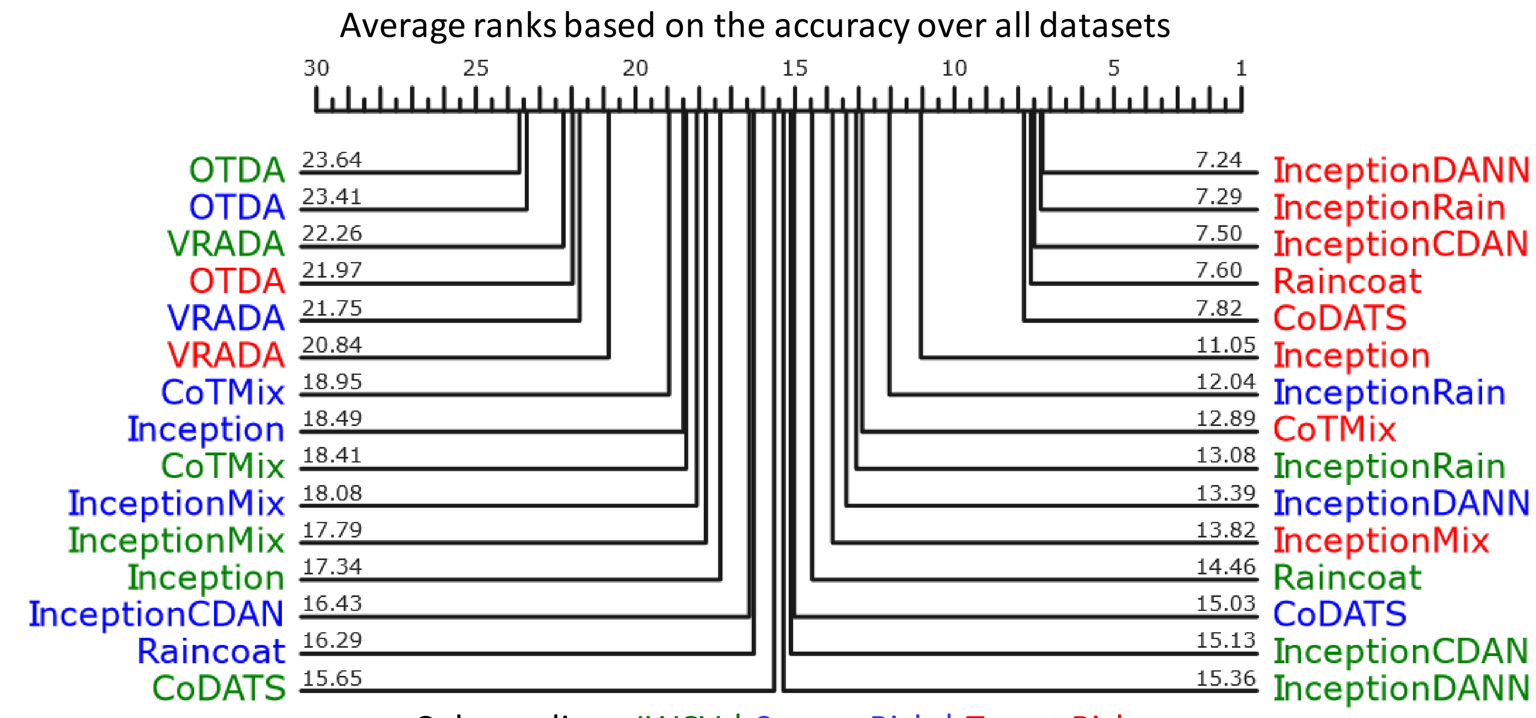
$$IWCV(f) = \frac{1}{n_S}\sum_{i=1}^{n_S}\frac{\hat{p}_T(X_i^S)}{\hat{p}_S(X_i^S)}\mathcal{L}(y_i^S, f(X_i^S))$$

Reference: https://arxiv.org/abs/2312.09857

Main Benchmark Results

Comparison of the different UDA approaches

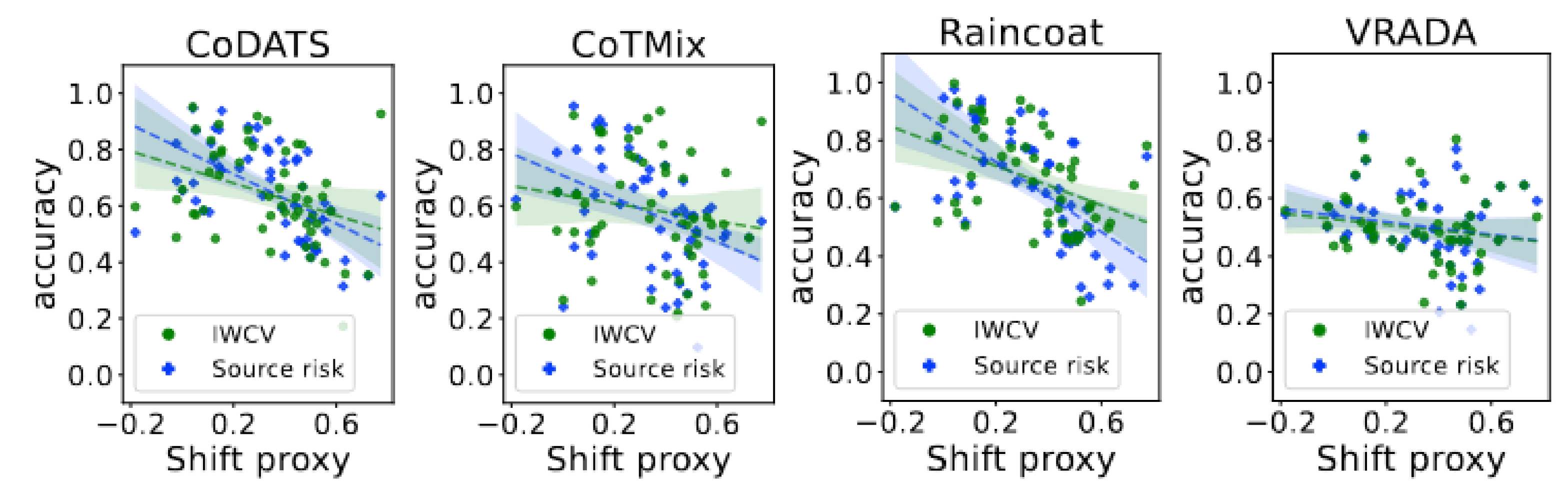
The different UDA approaches were tested on 12 different datasets from diverse themes (machinery, remote sensing, medical and human activity recognition). The best performance were obtained by **RainCoat** with Inception backbone (**InceptionRain**), both with IWCV and Source risk.



Color coding: IWCV | Source Risk | Target Risk

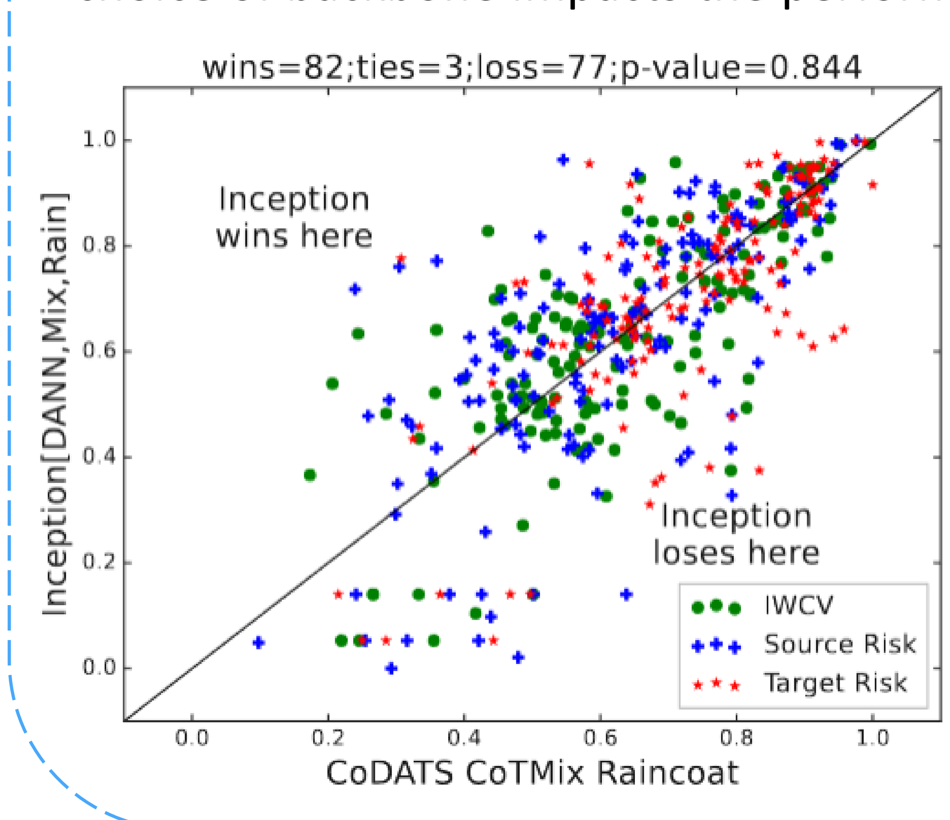
Impact of the shift on performance

We estimated the shift thought the relative difference in performance of a non-DA approach (InceptionTime) between target and source domains. As expected, the performance of most approaches are degraded when the shift increases. Interestingly, IWCV improves performance for high shifts, while Source risk is sufficient for low shifts.



Comparison of backbones

We ran a Wilcoxon signed-rank test to investigate whether the choice of backbone impacts the performance.



Even though Inception's backbone ranks higher, the improvement is not significant. It suggests that the main difference stems from the UDA technique itself rather than the choice of backbone.

Conclusion & Perspectives

- The best algorithm over 1458 experiments is **Raincoat**, which considers at the same time the temporal and the feature shift.
- **Inception backbone** enhances by little algorithms' performance.
- Our framework enables a fair comparison among algorithms by employing various hyperparameter tuning methods maintaining a consistent time budget for the tuning process.
- The larger the shift between domains, the greater the necessity for employing **hyperparameter techniques**.

➔ Can we estimate the shift (time & feature) for time series data?

Myriam Frikha, Hassan Ismail Fawaz, Ganesh del Grosso, Tanguy Kerdoncuff, Aurélie Boisbunon, Illyyne Saffar

Ericsson Research

*firstname.lastname@ericsson.com