DANNTe: a case study of a turbo-machinery sensor virtualization under domain shift

Baker Hughes
https://www.bakerhughes.com/

Luca Strazzera, Valentina Gori, Giacomo Veneri
TPS AI Team - Baker Hughes

NEURAL INFORMATION PROCESSING SYSTEMS

Introduction

Unsupervised Domain Adaptation, with generalization bounds stated by Ganin et al. [1,2,3] is a type of transfer learning where the task remains the same while the domains are different. The learner has access to a labeled source dataset and an unlabeled target dataset, where source and target datasets follow different probability distribution.

We seek to build a domain-invariant feature representation, to ensure good performance on both source and target domain. We apply the unsupervised DA method to an industrial turbo-machinery context providing practical results on a complex timeseries application, even in presence of a non independently and identically distributed assumption.

Use Case

The turbo-machinery application described in this work consists in building a virtual sensor using data collected from a prototype unit during winter-time and applying it to data collected from the same prototype, during summer-time. The domain shift we are facing is thus mainly related to different ambient conditions, influencing the distribution of the input features.

Dataset

The dataset used to validate the **DANNTe** approach is the collection of timeseries acquired from 30 sensors installed on a gas turbine prototype. Data collected in winter (from December to February) are used as labelled source dataset, while data collected in summer (from June to July) as unlabelled target dataset. In Fig. 1 the distributions (in source and target datasets) of some features inputting the model is shown, as examples.

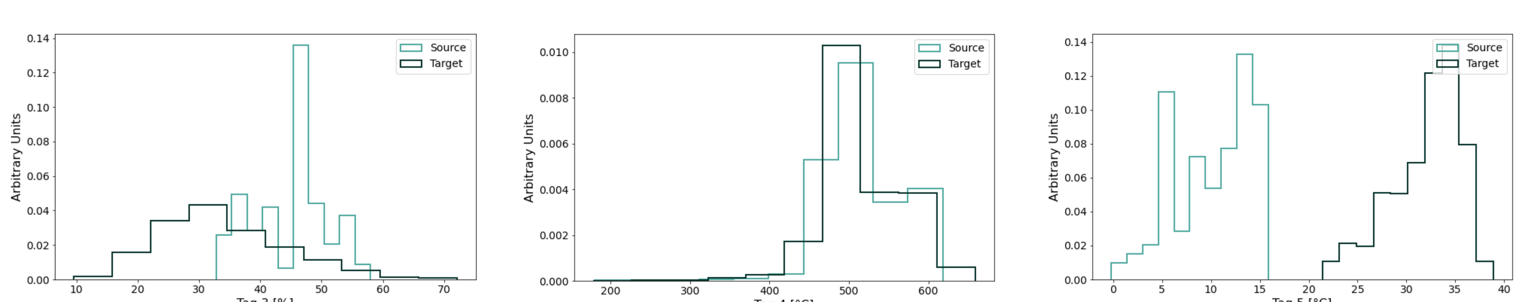
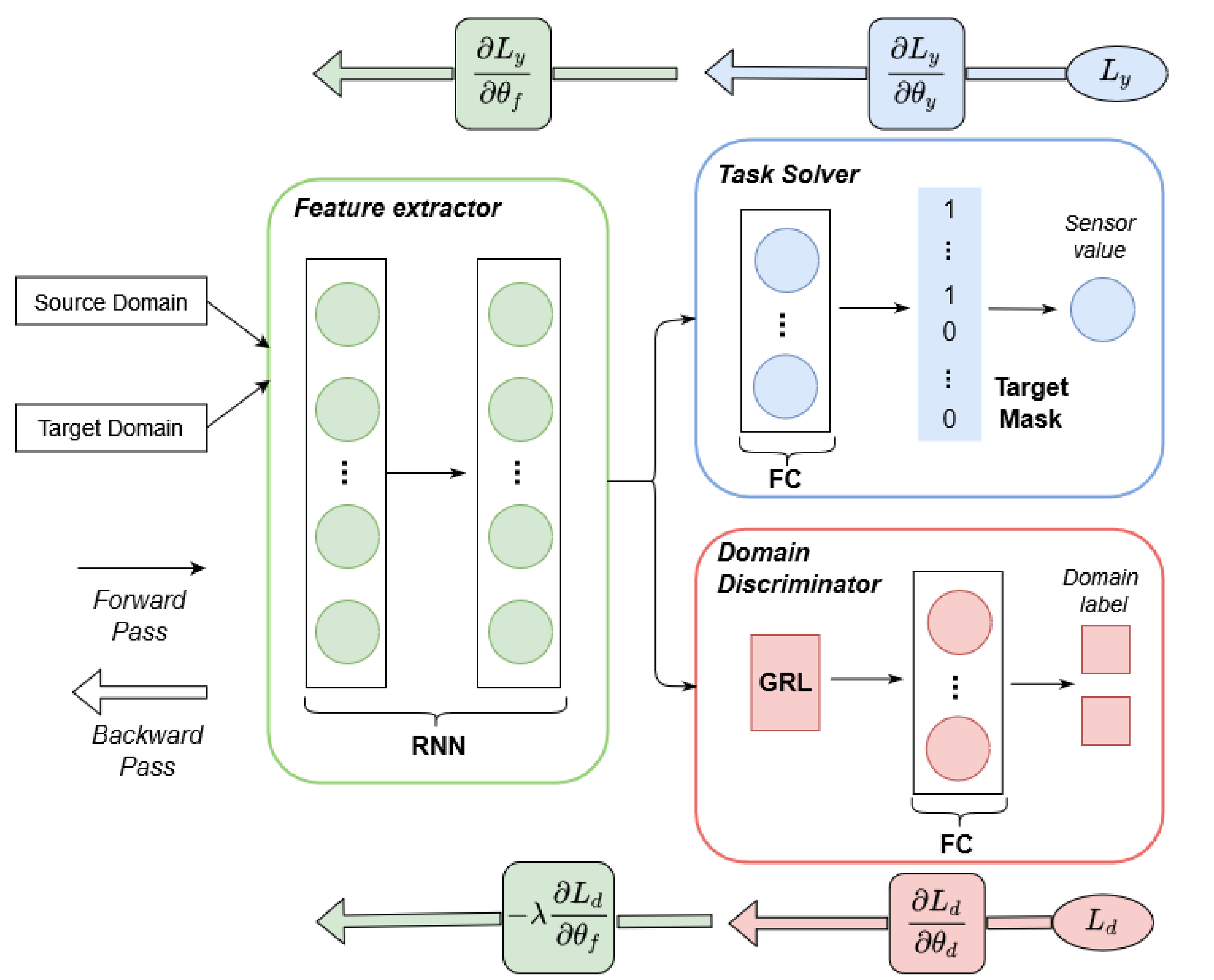


Fig 1. Distribution of some input features in source (light green) and target (dark green) dataset.

The implemented Architecture

Architecture of our proposed approach (DANNTe), based on Ganin et al. Feature extractor weights are modified by both the task solver (in our case, a regressor trying to minimize the reconstruction loss) and the domain classifier (trying to minimize the source vs target domain classification loss). The gradient reversal layer acts so that a minimization problem is solved (instead of a min-max one), just reversing the sign of the domain classifier gradient during backpropagation.



Since we are dealing with timeseries we propose to use a RNN (LSTM) as a feature extractor, to consider the time dependencies in our dataset. The solution we present is to train the model by creating equally divided batches where half of each batch is filled with samples from the source domain, and half with samples from the target domain, maintaining the time ordering.

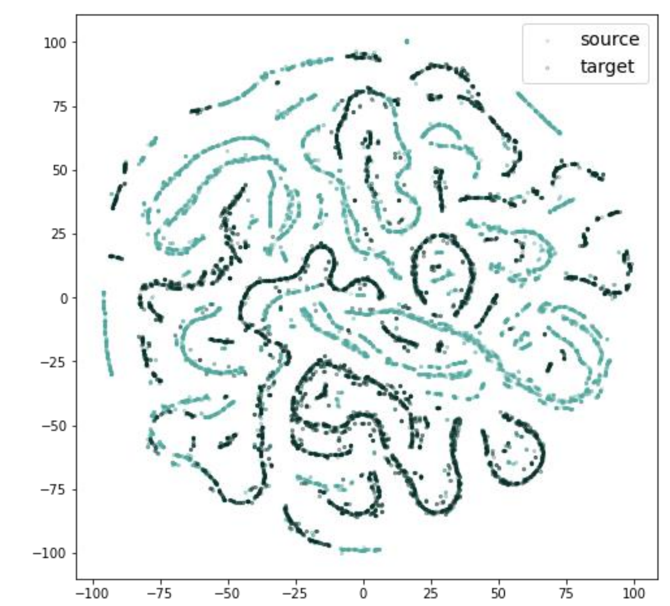
To reduce the training time we add a target mask layer to set the contribution of the target domain samples to zero in the task solver loss.

Results

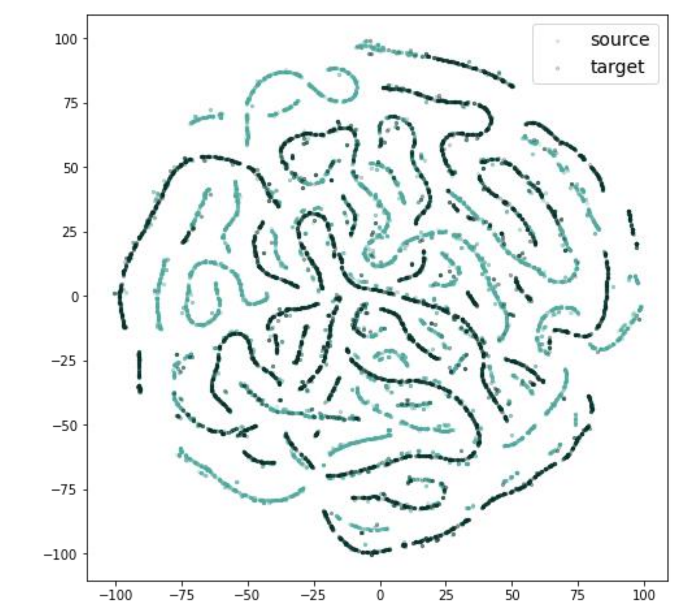
Model	MSE (Source)	MSE (Target)	MAPE (Target)	KL-divergence
Constant (mean)	$99.9_{\pm 0.0}$	$77.5_{\pm 0.0}$	12.0	-
Baseline	$\mathbf{0.7}_{\pm 0.1}$	$5.1_{\pm 0.9}$	3.4	6.1
Fully-supervised	$0.9_{\pm 0.7}$	$\mathbf{1.8}_{\pm 0.8}$	**0.9**	**1.2**
DANN	$0.8_{\pm 0.3}$	$3.6_{\pm 0.6}$	2.5	2.7
DANNTe	$0.8_{\pm 0.4}$	$2.3_{\pm 0.4}$	1.5	2.5

Embedding (dimensionality reduction by U-Map)

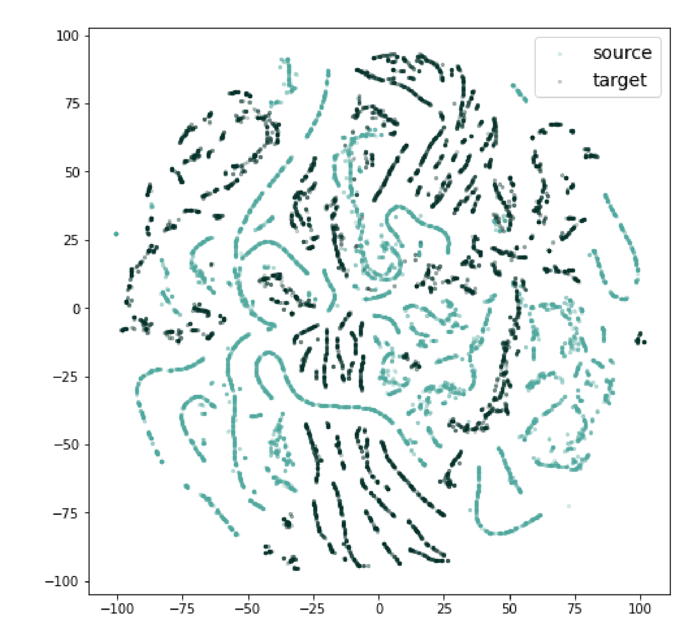
Two-dimensional data representation using UMAP. We can observe a clear separation between the two domains (light green and dark green, source and target respectively) using the baseline model (a). Both Fully Supervised (b) and DANNTe (c) construct a less discriminative feature representation.



(a) Baseline Model



(b) Fully supervised Model



(c) DANNTe

[1] Yaroslav Ganin et al. "Domain-adversarial training of neural networks". In:The journal of121machine learning research17.1 (2016), pp. 2096–2030
[2] Shai Ben-David et al. "Analysis of Representations for Domain Adaptation". In:Advances in114Neural Information Processing Systems 19 (NIPS 2006)(2007), pp. 137–144
[3]Shai Ben-David et al. "Impossibility Theorems for Domain Adaptation". In:Proceedings of the 116Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR 9(2010),117pp. 129–136