



A transfer-learning approach for corrosion prediction in pipeline infrastructures

Giuseppe Canonaco¹ · Manuel Roveri¹ · Cesare Alippi¹ · Fabrizio Podenzani² · Antonio Bennardo² · Marco Conti² · Nicola Mancini²

Accepted: 15 August 2021 / Published online: 6 October 2021
© The Author(s) 2021

Abstract

Pipeline infrastructures, carrying either gas or oil, are often affected by internal corrosion, which is a dangerous phenomenon that may cause threats to both the environment (due to potential leakages) and the human beings (due to accidents that may cause explosions in presence of gas leakages). For this reason, predictive mechanisms are needed to detect and address the corrosion phenomenon. Recently, we have seen a first attempt at leveraging Machine Learning (ML) techniques in this field thanks to their high ability in modeling highly complex phenomena. In order to rely on these techniques, we need a set of data, representing factors influencing the corrosion in a given pipeline, together with their related supervised information, measuring the corrosion level along the considered infrastructure profile. Unfortunately, it is not always possible to access supervised information for a given pipeline since measuring the corrosion is a costly and time-consuming operation. In this paper, we will address the problem of devising a ML-based predictive model for internal corrosion under the assumption that supervised information is unavailable for the pipeline of interest, while it is available for some other pipelines that can be leveraged through Transfer Learning (TL) to build the predictive model itself. We will cover all the methodological steps from data set creation to the usage of TL. The whole methodology will be experimentally validated on a set of real-world pipelines.

Keywords Corrosion prediction · Machine learning · Transfer learning · Covariate shift · Oil & Gas

1 Introduction

Internal corrosion in pipeline infrastructures used to transport gas and oil poses various threats to the environment and human beings in terms of both contamination and accidents, which may lead also to explosions in presence of gas leakages [1]. Therefore, the need to develop mechanisms able to predict the presence of such detrimental phenomenon is of utmost importance.

Unfortunately, this harmful phenomenon is very complex to be modeled or predicted. This is due to the incredible amount of factors it depends on. In carbon-steel pipelines, corrosion may be influenced by CO_2 , H_2S , oil or water

wetting, steel composition, and internal surface conditions [2]. Additionally, bacteria activity, fluid dynamics, transport conditions over the entire operating lifespan, as well as geometrical characteristics of the pipeline are certainly important influencing factors [3].

The complexity of the envisaged phenomenon is not the only critical issue for corrosion prediction. In fact, the difficulty in gathering information about all the influencing factors is an additional key concern. Indeed, only the geometrical characteristics of the pipeline and its fluid-dynamic information are available (the latter via simulation). On the other hand, information about all the other influencing factors is extremely difficult (or often impossible) to be gathered from the field. This is the reason why physical-based corrosion prediction models, e.g., [4], revealed to be not accurate in real-world corrosion scenarios.

Recently, Machine Learning (ML) techniques have proved to be able to model complex phenomena. Specifically, in our application scenario, there has been an increasing interest in leveraging these kinds of approaches in the context of

✉ Giuseppe Canonaco
giuseppe.canonaco@polimi.it

¹ Politecnico di Milano, Milan, Italy

² Eni S.p.A, Milan, Italy

internal corrosion modeling and prediction. For instance, in [3], a Neural Network (NN) is employed to build a predictive model of the corrosion phenomenon. Before training the NN, the most important features are selected with a Mutual Information-based approach. Similarly, a feature selection approach based on the sensitivity of the NN output w.r.t. infinitesimal variations of the input is applied in [5] before building a predictive model of internal corrosion, whereas in [6] the feature selection step is performed via a Grey Relational Analysis. One major critical point of these ML solutions is that, following a supervised approach, they require information about the presence of corrosion in a given pipeline. Such information is gathered directly from the field employing Pipeline Inspection Gauges (PIGs) that are robot inspection systems measuring the presence of corrosion within a pipeline. In other words, the main limitation of such supervised approaches for corrosion prediction is that a model is built only when information about the real presence of corrosion is already available (which is not usually the case for companies managing pipeline infrastructures).

Therefore, it is crucial to overcome such a limitation by designing corrosion prediction models able to operate even in the scenario where supervised information is not available for a given target pipeline. Achieving this goal would completely change the traditional way corrosion prediction is managed in oil and gas pipelines. Indeed, it would allow defining a predictive model of internal corrosion without sending a PIG through the target pipeline, hence avoiding such an expensive and time-consuming operation.

In this paper, to achieve this goal, we will leverage Transfer Learning (TL) to design an algorithmic solution for corrosion prediction when supervised information is not available for the target pipeline. TL allows overcoming the lack of supervised information on the target pipeline by exploiting a set of source pipelines where supervised information is available, which is a very common real-world scenario for oil and gas companies. In more detail, the proposed approach will rely on the joint use of a transductive TL technique and an Importance Weighted Cross-Validation (IWCV) technique; the former to build the predictive model onto the target pipeline and the latter to rank the sources w.r.t. their estimated performance over the pipeline of interest. Remarkably, the effectiveness of the proposed solution has been tested on a set of real-world gas pipelines.

The paper is structured as follows. In Section 2, an introduction to TL and the problem formulation are provided. In Section 3, our methodological approach is described in all its components, whereas, in Section 4 experimental results on a set of real-world pipelines are presented to thoroughly validate the proposed approach. Finally, in Sections 5 and 6, discussions and conclusions are drawn, respectively.

2 From supervised to transfer learning in corrosion prediction

From a machine learning perspective, predicting the presence of corrosion in an oil or gas pipeline requires effective modeling of the relationships between the corrosion-influencing factors and the presence of corrosion in a pipeline. More specifically, a pipeline is divided into bars, i.e., independent segments of the pipeline, and the machine learning task aims at modeling the relationship between the corrosion-description factors of each bar and its level of corrosion.

Let us define a learning task as the tuple $(X, P(x), Y, P(y|x))$, where Y is the label set, $P(y|x)$ is the conditional distribution, X is the feature space and $P(x)$ is the marginal distribution. Here x represents the feature vector associated with a bar and y its corrosion level. In the context of a learning task, the objective is to learn a function correctly predicting y when evaluated on x .

In this setting, two main problems arise. First, as commented in Section 1, not all the corrosion influencing factors are available. Hence the feature vector x comprises only geometrical and fluid dynamical information about a bar: the former is available given the pipeline on-the-field deployment, while the latter is provided by a fluid-dynamical simulator [7]. Second, as highlighted in Section 1, the supervised information, i.e., the presence or absence of corrosion in a bar, is provided through an inspection campaign where a PIG passes inside the pipeline collecting a set of corrosion level measurements together with their GPS coordinates along the pipeline profile. This poses a relevant compatibility issue since the geometrical characteristics, mainly represented by the inclination and curvature of the pipeline profile, are not directly compatible with the fluid dynamical descriptors creating a resolution mismatch between the information gathered through the PIG and the simulator [7]. More precisely, we can obtain fluid-dynamical data down to a certain granularity under which the simulator starts returning non-stationary solutions. This issue has been highlighted and addressed in [8] and we will follow this approach in the context of this work to build a given data set $D = \{(x_i, y_i)\}_{i=1}^n$ with x representing the feature vector, made of fluid dynamical and geometrical components, and y representing the corrosion level.

From the machine learning perspective, supervised learning techniques, like the ones mentioned in Section 1, work under the assumption that the distribution of training and test data is the same. In real-world oil and gas pipeline infrastructures, it may happen that we would like to solve a certain target task $(X_T, P_T(x), Y_T, P_T(y|x))$ for which some of the elements allowing a proper application of supervised learning techniques are missing (e.g., we do not have access to the labels since they are too costly to be

retrieved), but we can access another related source task $(X_S, P_S(x), Y_S, P_S(y|x))$ that may help us overcome the difficulties we would have in learning the target task by itself. This implies that we cannot directly build a predictive model onto a given pipeline $(X_S, P_S(x), Y_S, P_S(y|x))$ for which the supervised information is available and then reuse it onto another pipeline $(X_T, P_T(x), Y_T, P_T(y|x))$ for which the supervised information is missing. We emphasize that there is a potential distributional shift to take into account and our predictive models may suffer a performance hindering if this is not properly considered.

In this setting, TL allows us to leverage the source task in order to better approximate the conditional distribution of the target task. In the TL literature, several approaches are available differing in how source and target tasks relate to each other (see [9] for an extensive treatment). For instance, we can talk about *inductive* TL whenever $(Y_S, P_S(y|x)) \neq (Y_T, P_T(y|x))$ (notice that $(X_S, P_S(x)) = (X_T, P_T(x))$ or $(X_S, P_S(x)) \neq (X_T, P_T(x))$), whereas we have *transductive* TL whenever $(Y_S, P_S(y|x)) = (Y_T, P_T(y|x))$ and $(X_S, P_S(x)) \neq (X_T, P_T(x))$. The former requires access to at least a set of labeled data from the target context, whereas the latter only requires a set of unlabeled data coming from it. Furthermore, we can classify TL approaches w.r.t. the knowledge being transferred across the two tasks: parameters [10, 11], features [12–15], samples [16–18] or relational knowledge [19, 20]. Finally, TL techniques may be either homogeneous or heterogeneous as they are classified in [21, 22], where, in [22], we may find a review of recent works with the main focus on homogeneous approaches.

In the context of this work, we will focus on *transductive* TL techniques because companies managing pipelines usually have supervised information only for some of their pipeline infrastructures. In particular, this approach is justified by the assumption that, as commented by the domain experts, the corrosion phenomenon is the same in two different pipelines provided that the influencing factors are equal, which translates into $P_S(y|x) = P_T(y|x)$ (trivially, the label set Y does not change w.r.t. the considered pipeline). This implies that, in the context of corrosion prediction for pipeline infrastructures, we can assume that $(Y_S, P_S(y|x)) = (Y_T, P_T(y|x))$ and $(X_S, P_S(x)) \neq (X_T, P_T(x))$, where the change is given by $P_S(x) \neq P_T(x)$ since the feature space is the same.

3 The proposed transfer-learning approach for corrosion prediction

An overview of the proposed approach based on TL for corrosion prediction in pipeline infrastructures is given in Fig. 1. More specifically, we have a set of source pipelines

for which the supervised information is available and a target pipeline where only the influencing factors of the corrosion are accessible. Our goal is to create a predictive model for the target pipeline leveraging the source pipelines thanks to TL. This last step is done in two phases: at first, we rank the models built on a given source w.r.t. an estimate of their performance on the target (Algorithm 1 in Fig. 1); then the best B sources (being B a tunable parameter of the algorithm) are selected to build a predictive model for the target in a multi-task manner (Algorithm 2 in Fig. 1). Finally the predictive model $\hat{\theta}$ is used onto the target pipeline to get the corrosion levels.

In the rest of this section, we will initially discuss how to build a model for the target pipeline when only one source is available, then we will review how to estimate the performance of a model onto the target pipeline, and finally, we will show how to leverage more than one source to build the target model in a multi-task manner.

3.1 Transfer Learning from a source to a target pipeline

Given $D_S = \{(x_i^S, y_i^S)\}_{i=1}^n$ and $D_T = \{x_i^T\}_{i=1}^{n'}$ representing the source and target pipelines data, the problem we want to tackle is:

$$\theta^* \in \arg \min_{\theta \in \Theta} \mathbb{E}_{(x,y) \sim P_T(\cdot, \cdot)} [l(x, y, f(x, \theta))], \quad (1)$$

where θ is the vector parametrizing our predictor $f(x, \theta)$ and l is a fixed loss function measuring the mismatch between predictions and ground truth. We emphasize that samples coming from $P_T(x, y)$ are not available, but we have samples coming from $P_T(x)$ and $P_S(x, y)$. Therefore, by exploiting Importance Sampling (IS) [23], we can rewrite the objective of (1) in the following way:

$$\mathbb{E}_{(x,y) \sim P_S(\cdot, \cdot)} \left[\frac{P_T(x, y)}{P_S(x, y)} l(x, y, f(x, \theta)) \right]. \quad (2)$$

Now, since we have finite samples, we reformulate our problem by optimizing the empirical version of the above objective function:

$$\hat{\theta} \in \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n \frac{P_T(x_i^S, y_i^S)}{P_S(x_i^S, y_i^S)} l(x_i^S, y_i^S, f(x_i^S, \theta)), \quad (3)$$

which gives us a consistent estimator of $\hat{\theta}$ [24].

The problem of optimizing (3) can be tackled only if we can estimate the ratio $\frac{P_T(x,y)}{P_S(x,y)}$. Since we do not have labels coming from the target pipeline, the aforementioned estimation, in a general case, cannot be achieved. In our scenario, we know that, by assumption of the domain experts,

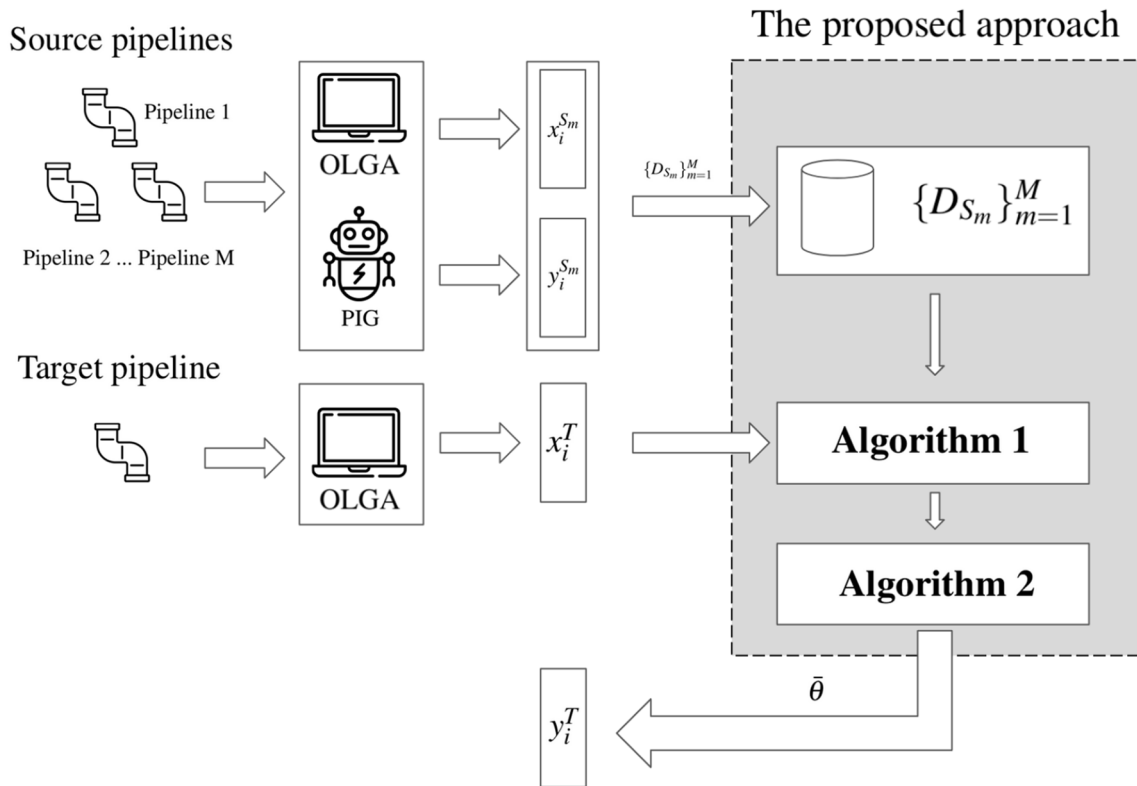


Fig. 1 Comprehensive scheme of the approach (this image has been designed using resources from Flaticon.com)

$P_S(y|x) = P_T(y|x)$. Hence, by definition of conditional probability, $\frac{P_T(x,y)}{P_S(x,y)} = \frac{P_T(x)}{P_S(x)}$, allowing an estimation of (3) on the available data.

In order to estimate the ratio $\omega(x) = \frac{P_T(x)}{P_S(x)}$, we will resort to the approach proposed in [18], which consists in solving the following Kernel Mean Matching (KMM) optimization problem:

$$\begin{aligned} & \min_{\omega} \|\mu(P_T) - \mathbb{E}_{x \sim P_S(\cdot)} [\omega(x)\Phi(x)]\|, \\ & \text{subject to :} \\ & \omega(x) \geq 0 \text{ and } \mathbb{E}_{x \sim P_S(\cdot)} [\omega(x)] = 1, \end{aligned} \tag{4}$$

where $\mu(P_T) = \mathbb{E}_{x \sim P_T(\cdot)} [\Phi(x)]$ and $\Phi : X \rightarrow \mathfrak{F}$ (being \mathfrak{F} a feature space). Then, under the assumptions that P_T is absolutely continuous w.r.t. P_S and \mathfrak{F} is a Reproducing Kernel Hilbert Space (RKHS) with universal kernel $k(x, u) = \langle \Phi(x), \Phi(u) \rangle$, we have that a solution to (4) is such that $P_T(x) = \omega(x)P_S(x)$. Unfortunately, both $\mu(P_T)$ and $P_S(x)$ are a priori unknown, but we have samples $\{x_i^S\}_{i=1}^n$ and $\{x_i^T\}_{i=1}^{n'}$ to rely on. Considering the empirical version of the objective function in (4), under the assumptions that $\omega : X \rightarrow [0, W]$ is a fixed function with finite mean and non-zero variance given $x_i^S \sim P_S$ (W is an upper bound on how much the two distributions can be different on a given $x \in X$), $\{x_i^T\}_{i=1}^{n'}$ is an iid (independent and identically distributed) set of samples

drawn from $P_T(x) = \omega(x)P_S(x)$ and $\|\Phi(x)\| \leq R$ for any $x \in X$ then with probability at least $1 - \delta$:

$$\left\| \frac{1}{n} \sum_{i=1}^n \omega(x_i^S) \Phi(x_i^S) - \frac{1}{n'} \sum_{i=1}^{n'} \Phi(x_i^T) \right\| \leq \left(1 + \sqrt{-\log\left(\frac{\delta}{2}\right)} \right) R \sqrt{\frac{W^2}{n} + \frac{1}{n'}}, \tag{5}$$

giving us an upper bound on the empirical optimization outcome (notice that the larger W , the larger the number of samples needed to obtain meaningful convergence guarantees). Now, letting $K_{i,j} = k(x_i^S, x_j^S)$, $\kappa_i = \frac{n'}{n} \sum_{j=1}^{n'} k(x_i^S, x_j^T)$ and $\bar{\omega}_i = \omega(x_i^S)$, the left hand side of (5) squared can be reformulated as follows:

$$\frac{1}{n^2} \bar{\omega}^T K \bar{\omega} - \frac{2}{n^2} \kappa^T \bar{\omega} + \text{const}, \tag{6}$$

which yields the following optimization problem:

$$\begin{aligned} & \min_{\bar{\omega}} \frac{1}{2} \bar{\omega}^T K \bar{\omega} - \kappa^T \bar{\omega} \\ & \text{subject to :} \\ & \bar{\omega}_i \in [0, W] \text{ and } \left| \sum_{i=1}^n \bar{\omega}_i - n \right| \leq n\epsilon. \end{aligned} \tag{7}$$

The first constraint in (7) provides an upper bound to the degree up to which the two distributions may be different, whereas the second one forces $\omega(x)P_S(x)$ to be close to a probability measure. For a more detailed discussion about KMM please refer to [18] and the references therein. Solving the optimization problem stated in (7) will return us the weight vector $\bar{\omega}$ to correct the shift in distribution between the source and target pipelines.

3.2 Ranking the source pipelines

In a real-world scenarios, we may have multiple available source pipelines, $\{D_{S_m}\}_{m=1}^M$. Hence, we would like to identify the most appropriate source within the given set. More precisely, we would like to select the source pipeline which allows us to minimize the generalization error defined as:

$$\mathbb{E}_{\{(x_i^T, y_i^T)\}_{i=1}^n, u, v} [l(u, v, f(u, \hat{\theta}))], \quad (8)$$

where (u, v) is a test point coming from the target pipeline and not present in the training set. The generalization error is usually estimated through cross-validation (CV), which, in this context, is useless because we do not have access to the target pipeline labels. However, we can obtain this estimate through k-fold Importance Weighted Cross-Validation (kIWCV) [25] as follows:

$$\frac{1}{k} \sum_{j=1}^k \frac{1}{|D_{S_m}^j|} \sum_{(x, y) \in D_{S_m}^j} \omega(x) l\left(x, y, f\left(x, \hat{\theta}_{D_{S_m}^j}\right)\right), \quad (9)$$

where $\hat{\theta}_{D_{S_m}^j}$ is the parametrization learned over the data set $D_{S_m} \setminus D_{S_m}^j$ (\setminus is the set difference operator) and the estimation of $\omega(x)$ can be performed by solving (7). The generalization error estimate provided by kIWCV with $k = n$ is almost unbiased and a similar claim can be proved for $k < n$ with a larger bias than that incurred with $k = n$ (for a more thorough treatment see [25]). Therefore, choosing the source pipeline S_m maximizing the above equation will allow us to get the best performance for our models on the target pipeline. Moreover, from a more general standpoint, (9) allows us to optimize all the hyperparameters of our model to maximize performance on the target pipeline.

In Algorithm 1, we combine KMM, used to estimate $\omega(x)$, and kIWCV. More specifically, at line 4, we compute the importance weights to correct the distributional shift between the current source and the target pipeline. At line 5, we compute the optimal hyper-parameters and their performance through kIWCV. At line 6, the performance and hyperparameters of the current source pipeline are appended to their respective lists, (i.e., $Perf$ and ρ). Finally, in line 8, we return the lists of performances and the

corresponding hyper-parameters. This completes the TL framework we will use in the context of corrosion prediction for pipeline infrastructures.

We could extend this solution by considering more than one source pipeline for the corrosion prediction. This can be done through a multi-task learning approach which could use up to all the source pipelines together with their related importance weights to build a model for the target one [26, Chapter 9]. However, evaluating the performance of all the possible subsets of source pipelines does not scale well. To avoid this issue we could combine the B best sources according to the ranking we obtain by sorting the results of Algorithm 1 (the effect of B will be experimentally evaluated in Section 4). This procedure is reported in Algorithm 2, where, at lines 4 and 5, we concatenate the data sets and the weights. At line 7, we compute the optimal hyperparameters of the learning algorithm, and, finally, at line 8, the best parametrization for the predictor is computed and subsequently returned.

Algorithm 1 Sources performance evaluation.

```

1: Input: Source data sets  $\{D_{S_m}\}_{m=1}^M$ , Target samples  $\{x_i^T\}_{i=1}^{n'}$ ,  $W, \epsilon, k$ 
2:  $Perf = [], \rho = []$ 
3: for  $m$  in  $1 \dots M$  do
4:    $\bar{\omega} = KMM(D_{S_m}, \{x_i^T\}_{i=1}^{n'}, W, \epsilon)$  solving (7)
5:    $hyperParams, p = IWCV(D_{S_m}, \bar{\omega}, k)$ 
6:    $Perf = Perf + [p], \rho = \rho + [hyperParams]$ 
7: end for
8: return  $\rho, Perf$ 

```

Algorithm 2 Multi-task TL

```

1: Input: The  $B$  best Source data sets obtained by Algorithm 1  $\{D_{S_m}\}_{m=1}^B$ , Target samples  $\{x_i^T\}_{i=1}^{n'}$ ,  $W, \epsilon, k$ 
2:  $D = [], \Omega = []$ 
3: for  $m$  in  $1 \dots B$  do
4:    $\Omega = \Omega + [KMM(D_{S_m}, \{x_i^T\}_{i=1}^{n'}, W, \epsilon)]$  solving (7)
5:    $D = D + [D_{S_m}]$ 
6: end for
7:  $hyperParams, Perf = IWCV(D, \Omega, k)$ 
8:  $\bar{\theta} = TrainLearningAlgorithm(D, hyperParams, \Omega)$ 
9: return  $\bar{\theta}$ 

```

4 Experiments

This section aims to evaluate the effectiveness of the proposed TL approach in real-world scenarios of corrosion prediction within 4 different gas pipeline infrastructures, namely P1, P2, P3, and P4. The section is organized as

Table 1 Geometrical variables

Geometrical variable	Meaning
<i>Odometry</i>	Odometry of a given bar
<i>Bar_{init}</i>	Odometric point where the bar begins
<i>Bar_{end}</i>	Odometric point where the bar ends
<i>Lat</i>	Latitude of the bar
<i>Long</i>	Longitude of the bar
<i>Elev</i>	Elevation of the bar
<i>Bar length</i>	Length of the bar

follows: Section 4.1 describes the employed data sets, whereas the experimental results are given in Section 4.2.

4.1 Data sets description

In order to integrate the geometrical and fluid dynamical data, we rely on the approach proposed in [8]. Such an integration approach will return us a data set $D = \{x_i, y_i\}_{i=1}^n$ for each pipeline. Here the vector x_i 's components represent the corrosion influencing factors (see Tables 1 and 2), whereas y_i is the scalar value representing the corrosion level (in terms of bar thickness percentage being corroded). Such a value is transformed into a categorical variable with four possible classes according to the thresholds provided by the domain experts and reported in Table 3. After the integration, a data transformation and enrichment is performed as follows. The *Lat*, *Long* and *Elev* are transformed into their equivalent Cartesian coordinates x , y and z . Then their first and second-order derivatives are computed (and

Table 2 Fluid-dynamical variables

OLGA Var.	Meaning	Unit
<i>ID_{flow}</i>	flow regime	-
<i>PT</i>	pressure	<i>bara</i>
<i>GT</i>	total mass flow rate	<i>kg/s</i>
<i>QT</i>	total volumetric flow rate	<i>m³/s</i>
<i>GG</i>	gas mass flow rate	<i>kg/s</i>
<i>QG</i>	gas volumetric flow rate	<i>m³/s</i>
<i>UG</i>	gas velocity	<i>m/s</i>
<i>TAUWG</i>	gas wall shear stress	<i>Pa</i>
<i>GLWVT</i>	total water mass flow rate including vapor	<i>kg/s</i>
<i>QLTWT</i>	water volumetric flow rate	<i>m³/s</i>
<i>UWTCNT</i>	water continuous velocity	<i>m/s</i>
<i>TAUWWT</i>	water film wall shear stress	<i>Pa</i>
<i>HOLWT</i>	water hold-up	-
<i>INCL</i>	horizontal inclination	<i>degree</i>
<i>TM</i>	temperature	<i>Celsius</i>
<i>A</i>	section area	<i>m²</i>

Table 3 Categories and Thresholds of the thickness percentage being corroded for a bar

Threshold	Category
$y < 0.03$	Absent
$0.03 \leq y < 0.08$	Low
$0.08 \leq y < 0.3$	Medium
$y \geq 0.3$	High

added to the feature set), which represent the pipeline profile's rate of change along a fixed component. Additionally, some features have been removed from the data sets because, according to the domain experts, they showed unexpected behavior, whereas some others were removed due to redundancy. More specifically, *GLWVT*, *QT*, *GT*, *QG*, *Bar length* and *GG* have been removed under domain expert advice, together with *Bar_{init}* because redundant. Finally, under the domain experts' suggestion, the samples associated with the beginning or end of the pipeline were removed because they have completely different behavior in terms of corrosion. At the end of this enrichment step, the features are normalized. Some summary information on the different pipelines after the above-mentioned transformations is provided in Table 4, where we may notice a severe imbalance among the various classes. It is worth noting that the High class is very rare and the Absent class is dominating the others.

4.2 Results

For each available target pipeline, we will evaluate the performance of Algorithm 2 in three different configurations: $B = 1$, $B = 2$ and $B = 3$. We used a Support Vector Machine (SVM) [27] with Radial Basis Kernel as $f(x, \theta)$. In Table 5, we report the results returned by Algorithm 1 when the F1-Score is chosen to perform the 10-IWCV (the model hyper-parameters are not reported for the sake of brevity). We will review each one of the aforementioned configurations of Algorithm 2 one target pipeline at a time. We will first look at the multi-class confusion matrices, then at their binarized version (reported in Appendix A) to check also the corrosion detection capabilities of each solution, and, finally, at the performances in F1-Score and accuracy.

Table 4 Label (y) distribution across the different pipelines

Pipeline	Data points	Low	Medium	High
P1	861	36	186	53
P2	3321	144	609	19
P3	1215	206	213	3
P4	1681	276	568	33

Table 5 Estimate of the performance (F1-Score) on the target pipeline through Algorithm 1

Target	1 st Source	2 nd Source	3 rd Source
P1	P3 0.56	P2 0.56	P4 0.45
P2	P1 0.71	P3 0.60	P4 0.45
P3	P1 0.71	P4 0.45	P2 0.30
P4	P1 0.71	P3 0.58	P2 0.56

4.2.1 P1

As we can see by comparing Fig. 2a and d, applying the TL technique gives us an improvement on the class low and the class absent recognition with a degradation effect on the class high and medium. For what concern the solution offered by using the first two sources ($B = 2$) according to IWCV (compare Fig. 2b and e), we have an improvement on both the absent and medium-class recognition with a degradation on the class low (the class high reduction is very slight). Using just the first ranked solution ($B = 1$) w.r.t. IWCV, we obtain a model that nearly always predicts absent (see Fig. 2f). If we take a look at the binarized version of the confusion matrices (see Fig. 6 in Appendix A), then only the

solution proposed by leveraging the first two sources w.r.t. IWCV is useful in terms of corrosion detection (compare Fig. 6a, b, c with d, e, f respectively). Finally, by looking at Tables 6 and 7, we may notice that applying transfer almost always increases our performance w.r.t. these two figures of merit.

4.2.2 P2

If we compare Fig. 3a and d ($B = 3$), we may notice an improvement on the absent class recognition at the expense of a degradation for the classes low and medium. For what concern Fig. 3b and e ($B = 2$), we can see an improvement on the recognition of the classes low and absent to which corresponds a slight reduction on the medium and high classes. By comparing Fig. 3c and f ($B = 1$), instead, we may see an improvement in the medium class recognition accompanied by a worsening on the absent class. Now, taking a look at the binarized versions of the confusion matrices (please compare Fig. 7a, b, c with d, e, f respectively in Appendix A) only the confusion matrix associated to the usage of the best source ($B = 1$) according to IWCV is meaningful. Finally, by looking at Tables 6 and 7, we see that applying transfer improves the accuracy and the F1-Score in all the

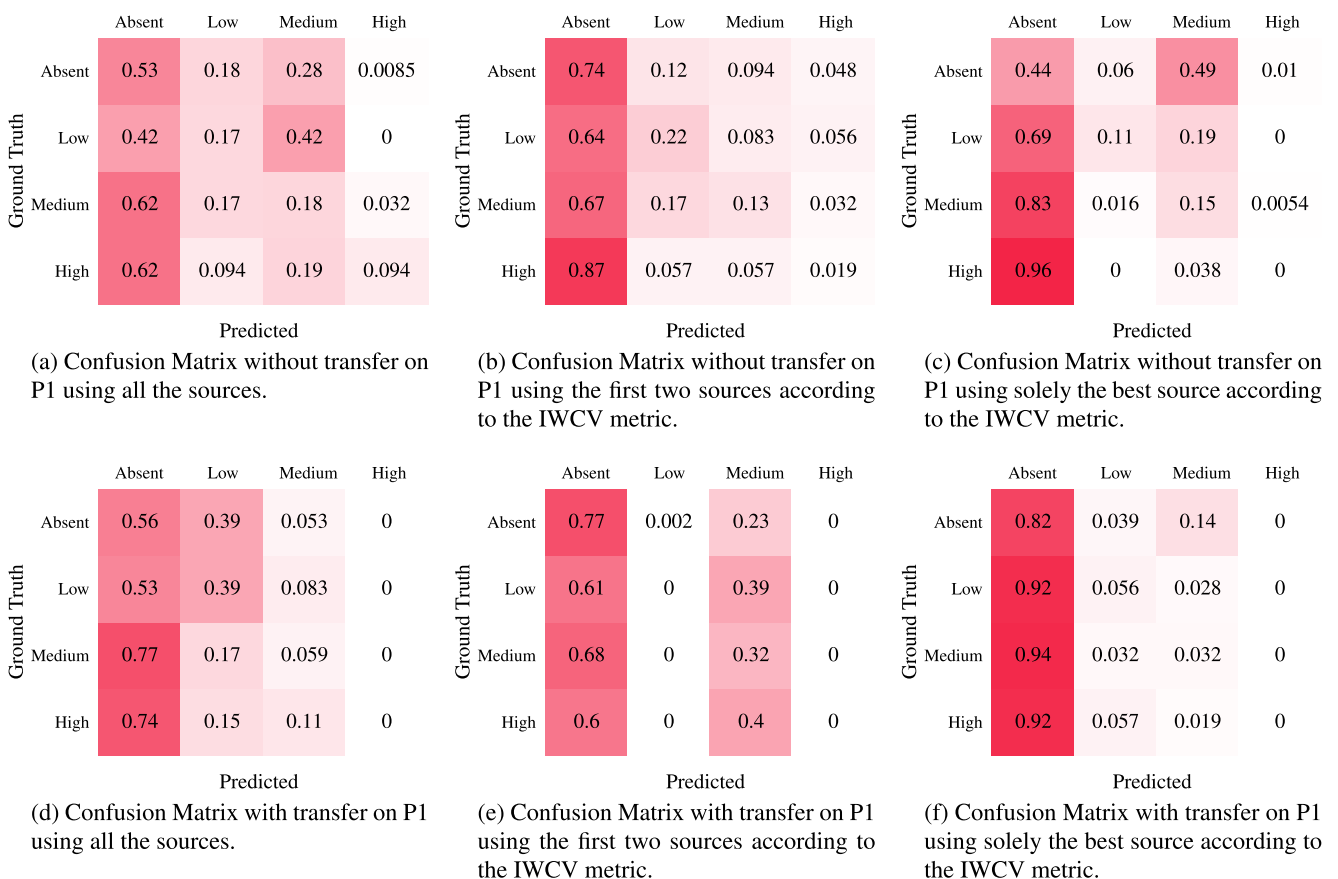
**Fig. 2** Multi-Class Confusion Matrices for P1

Table 6 Accuracy on target

Target	Supervised Oracle	First Two Sources $B = 2$		All $B = 3$		1^{st} Source $B = 1$	
		No TL	TL	No TL	TL	No TL	TL
P1	0.69	0.54	0.59	0.41	0.41	0.33	0.57
P2	0.50	0.54	0.58	0.35	0.42	0.71	0.68
P3	0.52	0.26	0.43	0.44	0.45	0.40	0.47
P4	0.38	0.27	0.41	0.35	0.44	0.41	0.43

The Supervised Oracle column is obtained by training an SVM onto a portion of the data set representing this target pipeline and then testing this model onto the held out part

Table 7 F1-Score on target

Target	Supervised oracle	First Two Sources $B = 2$		All $B = 3$		1^{st} Source $B = 1$	
		No TL	TL	No TL	TL	No TL	TL
P1	0.68	0.53	0.56	0.45	0.42	0.35	0.51
P2	0.57	0.57	0.60	0.41	0.48	0.70	0.68
P3	0.54	0.28	0.45	0.48	0.44	0.43	0.47
P4	0.41	0.30	0.38	0.35	0.41	0.35	0.36

The Supervised Oracle column is obtained by training an SVM onto a portion of the data set representing this target pipeline and then testing this model onto the held out part

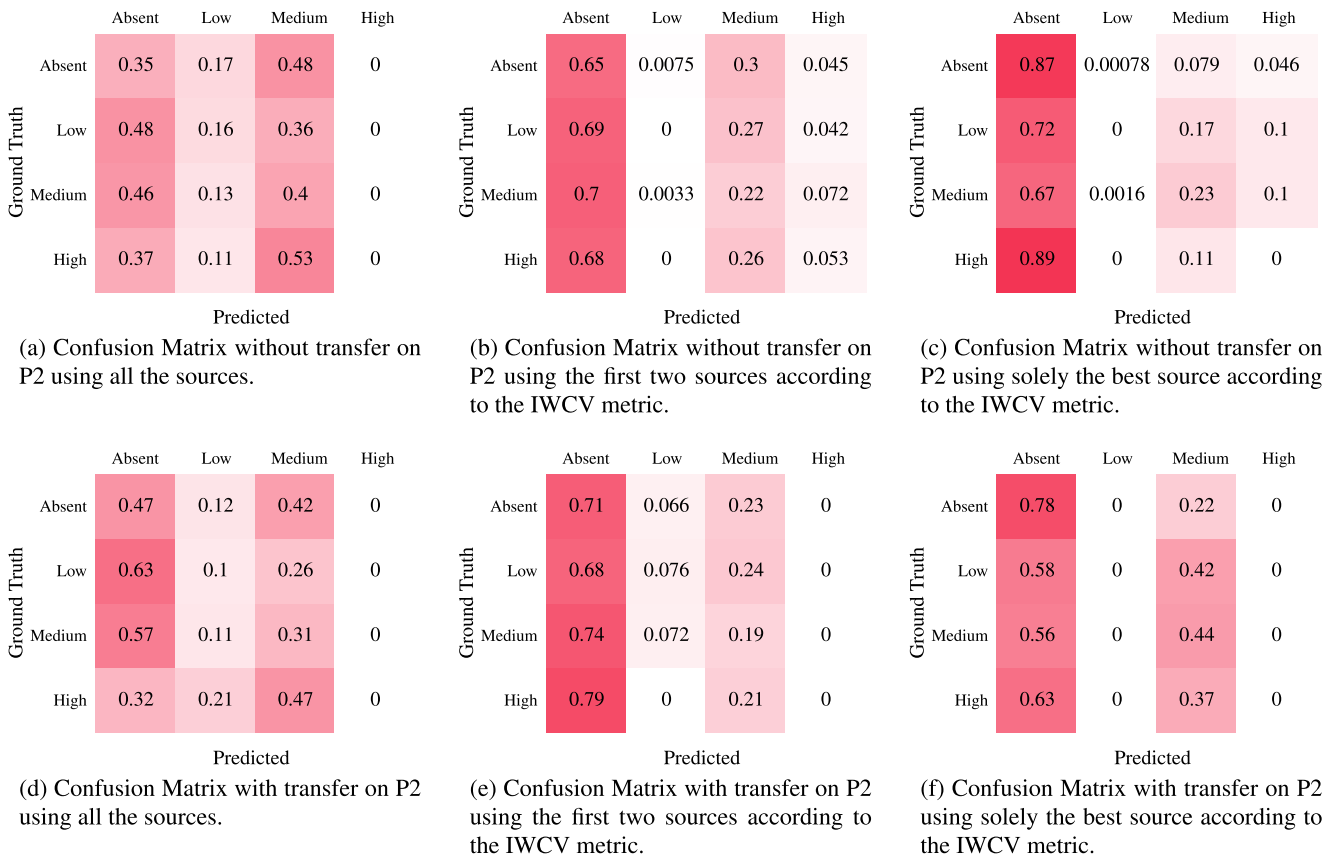


Fig. 3 Multi-Class Confusion Matrices on P2

cases except when we choose the best source ($B = 1$) w.r.t. IWCV. Furthermore, notice that, in some cases, the TL technique is able to get an improvement in F1-Score or accuracy w.r.t. to what was obtained by training a model onto a portion of the data set representing this target pipeline and then testing this model onto the held out part (i.e., the Supervised Oracle).

4.2.3 P3

Comparing Fig. 4a with d ($B = 3$) and b with e ($B = 2$), we may see as the TL technique improves the absent and medium classes recognition at the cost of increasing misclassifications within the low class. Differently, when comparing Fig. 4c with f ($B = 1$), we have the same behavior as before with also an increment on the misclassification error for the class high. In the context of the binarized versions (please see Fig. 8a, b, c compared with d, e, f, respectively, in Appendix A), we have that both the confusion matrices represented in Fig. 8e and f are meaningful. Also in the case of this target pipeline, we have that applying transfer almost always improves accuracy and F1-Score (see Tables 6 and 7).

4.2.4 P4

As we can see from Fig. 5a compared with 5d ($B = 3$) and Fig. 5b compared with 5e ($B = 2$), applying the TL technique allows us to improve the recognition performance both on the absent and medium classes with a degradation effect on the low class (in Fig. 5d a slight degradation on the high class is perceivable w.r.t. Figure 5a). In the context of Fig. 5c and f ($B = 1$), we have the same behavior described for Fig. 5a and d. Here, the only difference is that the absent class true positive rate remains the same. For what concern the binarized version (see Figure 9 in Appendix A), in the context of this target pipeline, choosing the best source ($B = 1$) w.r.t. IWCV does not produce a meaningful confusion matrix, whereas the other two approaches do (please compare Fig. 9a, b, c with d, e, f, respectively). Finally, as we may see from Tables 6 and 7, the TL technique always improves the two figures of merit surpassing or matching the performances obtained by training a model onto a portion of the target pipeline and then testing it onto the held out test set (i.e., the Supervised Oracle).

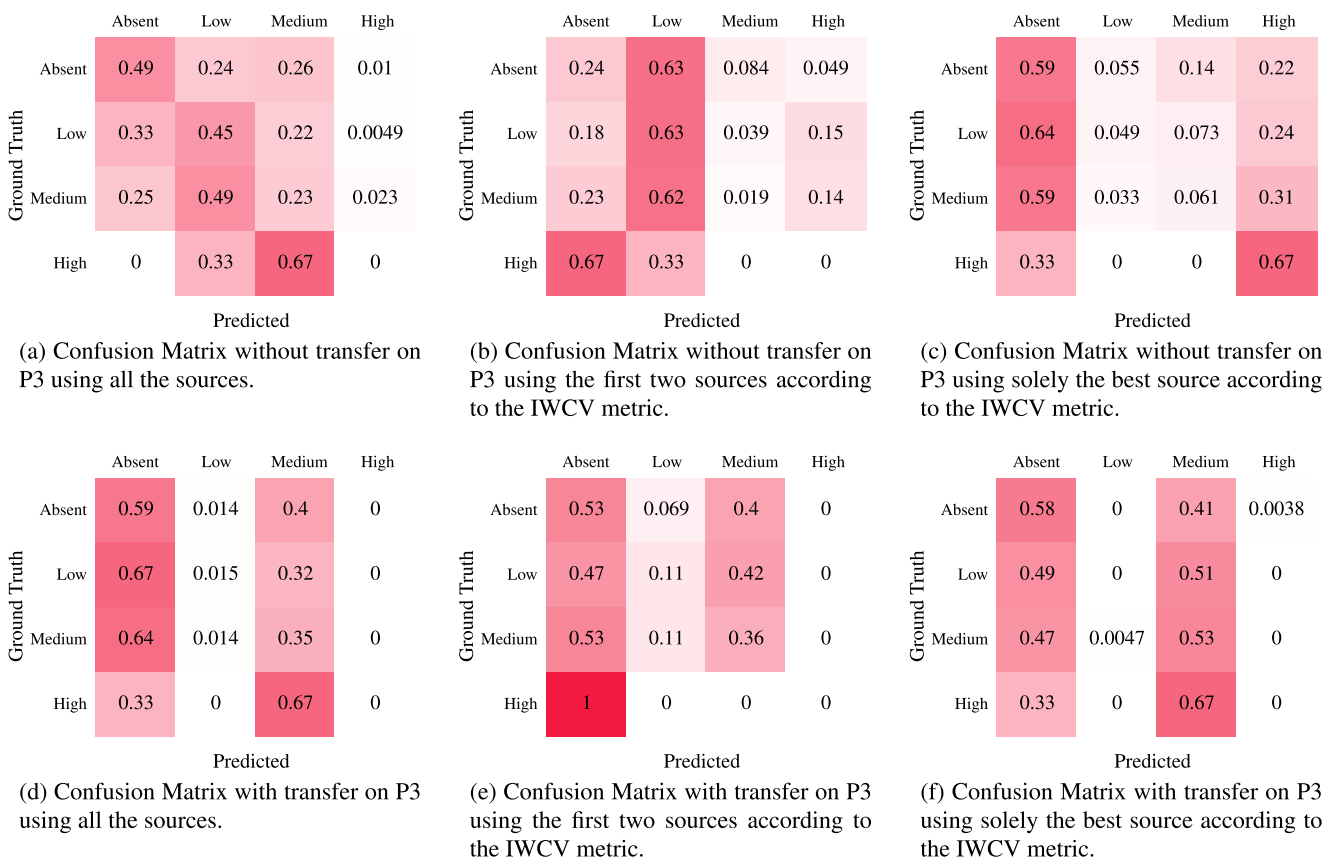


Fig. 4 Multi-Class Confusion Matrices on P3

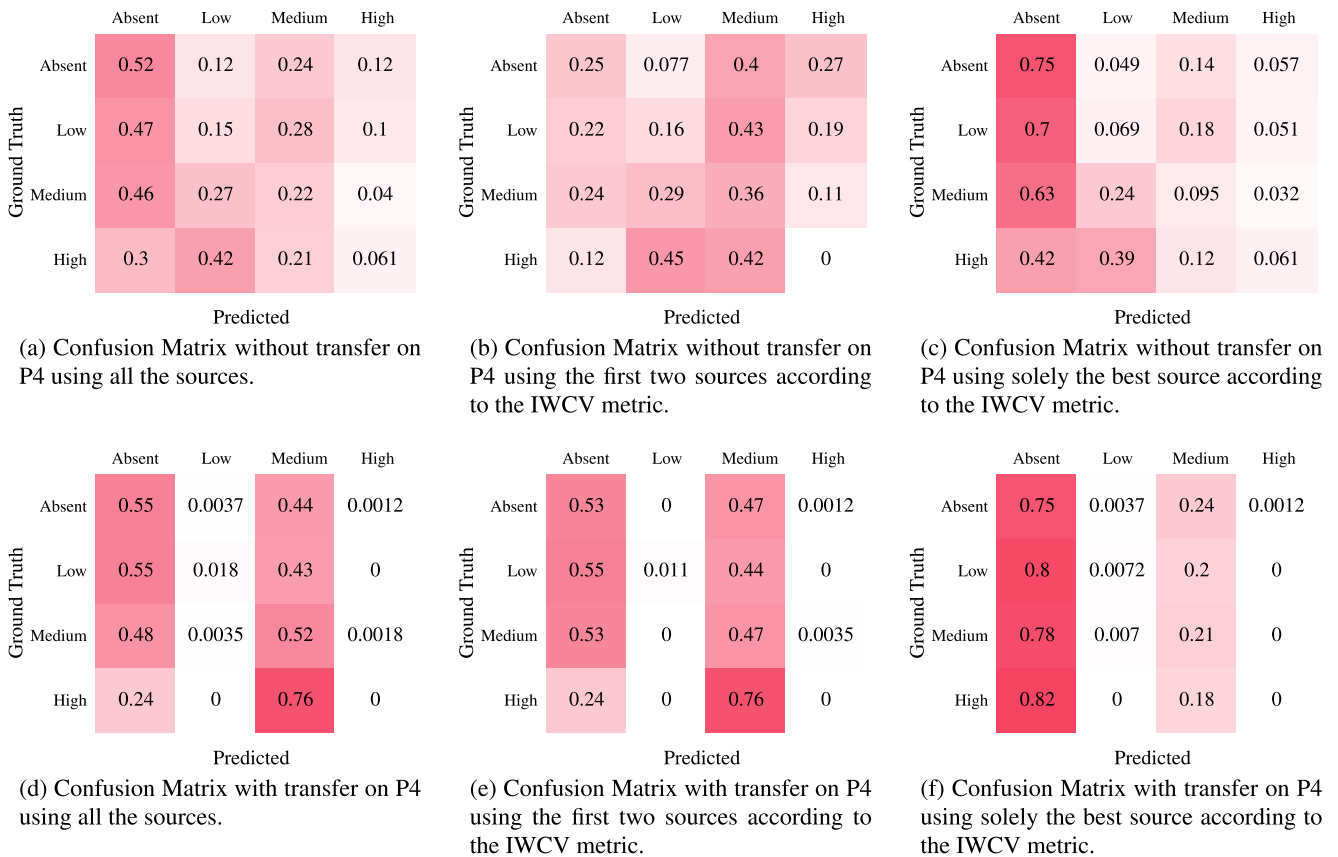


Fig. 5 Multi-Class Confusion Matrices on P4

5 Discussion

In the previous section, we have seen how the three configurations ($B = 1, 2, 3$) of the proposed solution behaved in terms of corrosion classification onto a target pipeline for which the labels are not available. The lack of labels was bridged with the aid of a TL technique described in Section 3.1. We have shown models' performance along four distinct components: the multi-class confusion matrices, their binarized version (to assess the detection capabilities of the developed models), the accuracy, and the F1-Score. Among the three different configurations, the one built using the first two best sources ($B = 2$) according to IWCV seemed to be the most stable, especially concerning the binarized version of the confusion matrices. This is a reasonable solution to trade off the intrinsic uncertainty in identifying the best source pipeline with the need to limit the number of sources to be used.

In the context of corrosion classification for oil and gas pipelines (the tackled application scenario), it would be also useful to get access to the confidence with which the model is predicting a certain label. To achieve this, for

what concerns SVMs, we can capture the probabilities for each class given a fixed sample. More precisely, multiclass probability estimates are computed by combining all pairwise probability estimates $r_{i,j}$ for class i and j . So, given all the $r_{i,j}$, the estimate of $p(y = i|x)$ is obtained by solving a linear system, see [28] for further details. The pairwise probability estimates are obtained by logistic regression on the score of the SVM [29]. It is noteworthy to point out that this technique to produce probabilities can be seamlessly integrated into the proposed TL approach.

6 Conclusion

In this paper, we tackled the problem of building a predictive model for the corrosion phenomenon in the context of a pipeline infrastructure for which the supervised information is not available. To achieve this goal we used a KMM-based TL technique to leverage labeled data coming from a source pipeline infrastructure. Moreover, in a context where different labeled source pipelines are available, we combined KMM, IWCV, and multi-task learning to produce

a model for the target infrastructure by selecting the appropriate sources.

As possible future directions, we would like to mention the need to acquire time-varying data for a set of fixed pipelines to develop models accounting for time variations of the corrosion phenomenon both in the TL context and in the supervised learning one. Furthermore, taking into account the pipelines' material and chemical composition of the blend flowing within the pipelines is another important future direction. As of right now, the material is homogeneous among our pipelines and we do not possess data describing the chemical composition of the blend flowing through them. However, it would be relevant to extend our methodology in this direction, maybe giving

more importance to those source pipelines having similar material and chemical composition of the blend w.r.t. the target pipeline. Finally, we hope that this work may stimulate a renewed interest in this complex application, maybe attracting more investments in research from the oil and gas companies.

Appendix A: Corrosion detection experimental results

In this section are reported all the binarized versions of the confusion matrices presented in Section 4. This is done to check the detection capabilities of the various models.

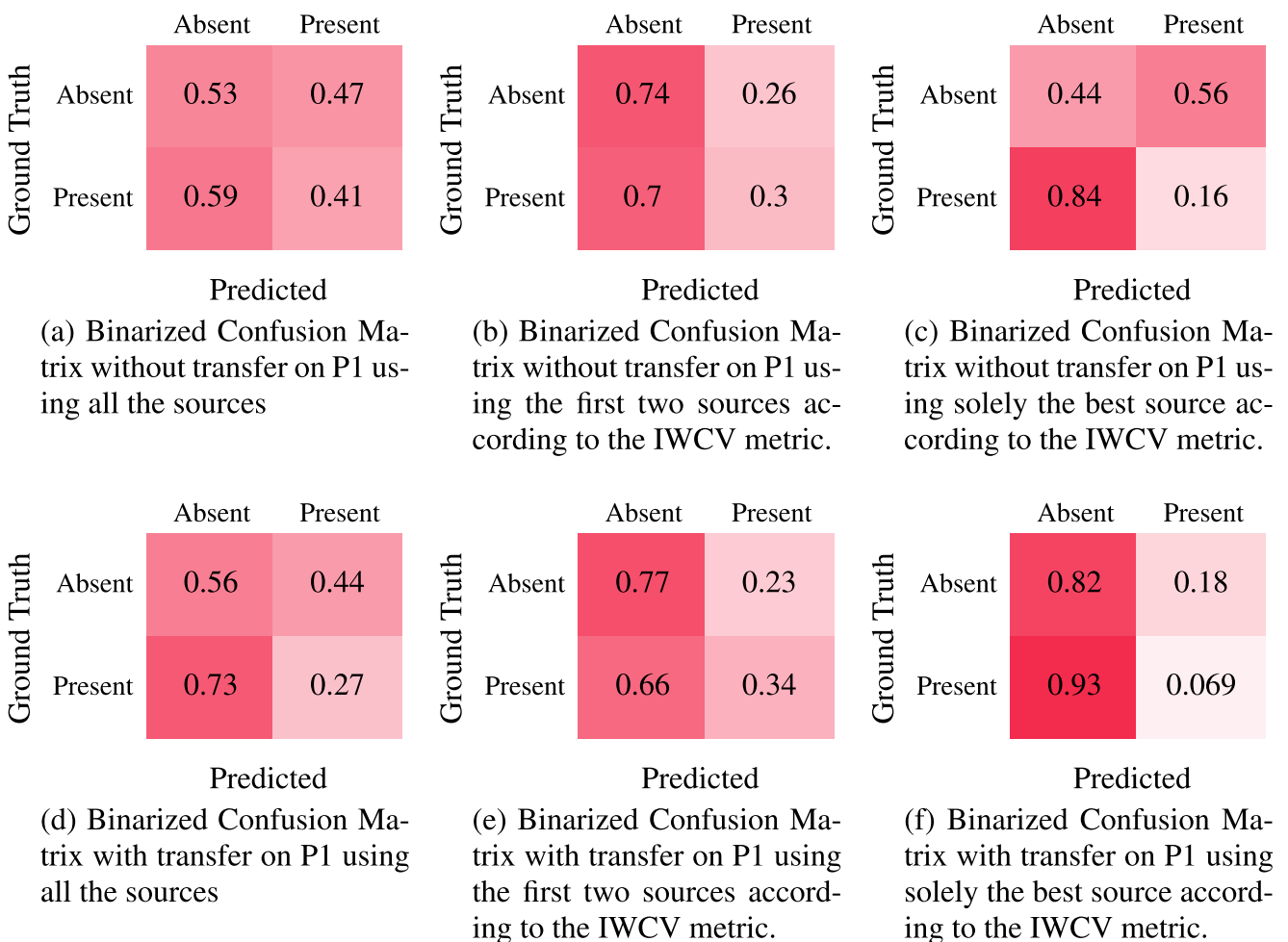


Fig. 6 Binarized Confusion-Matrices on P1

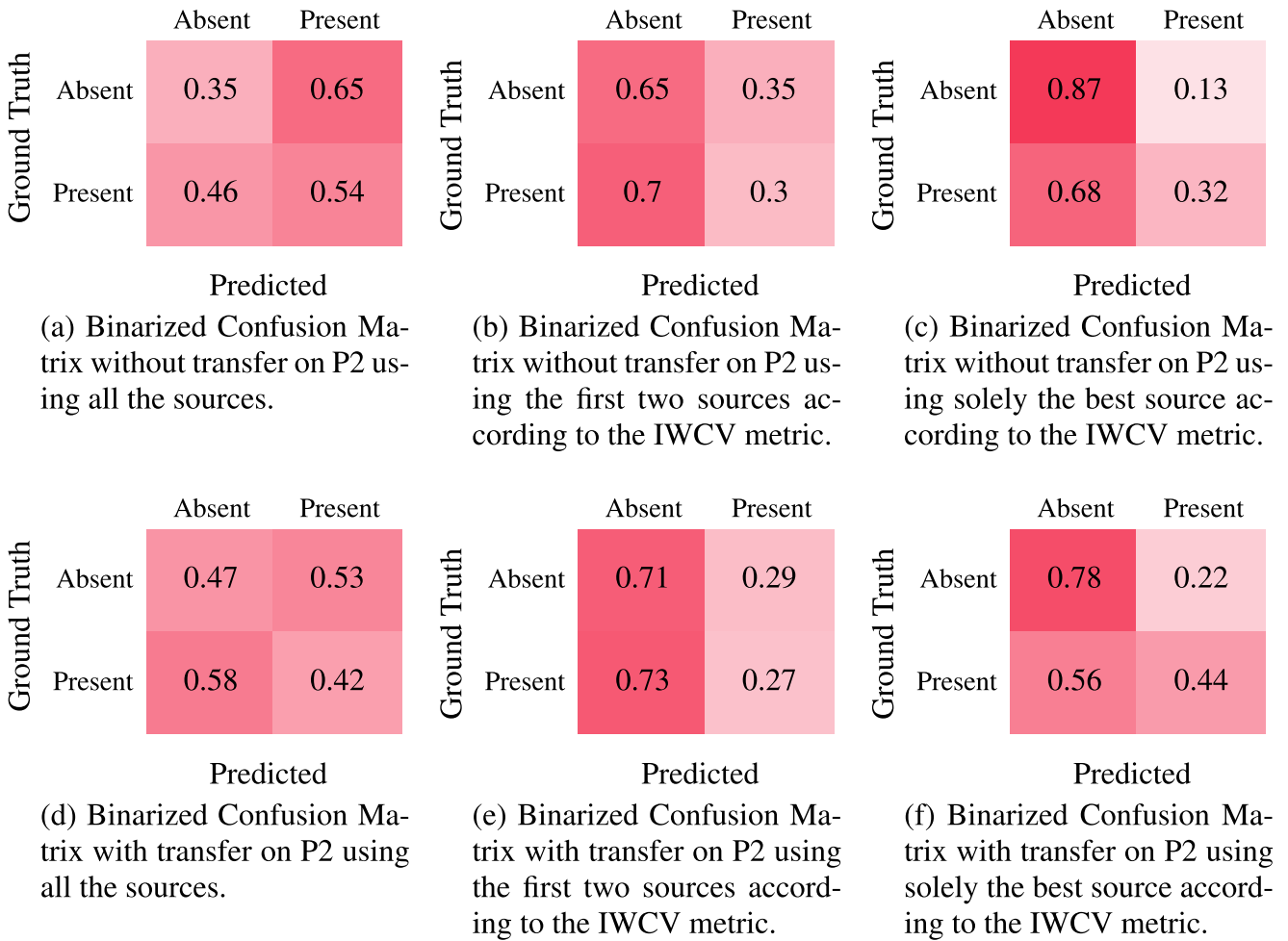


Fig. 7 Binarized Confusion Matrices on P2

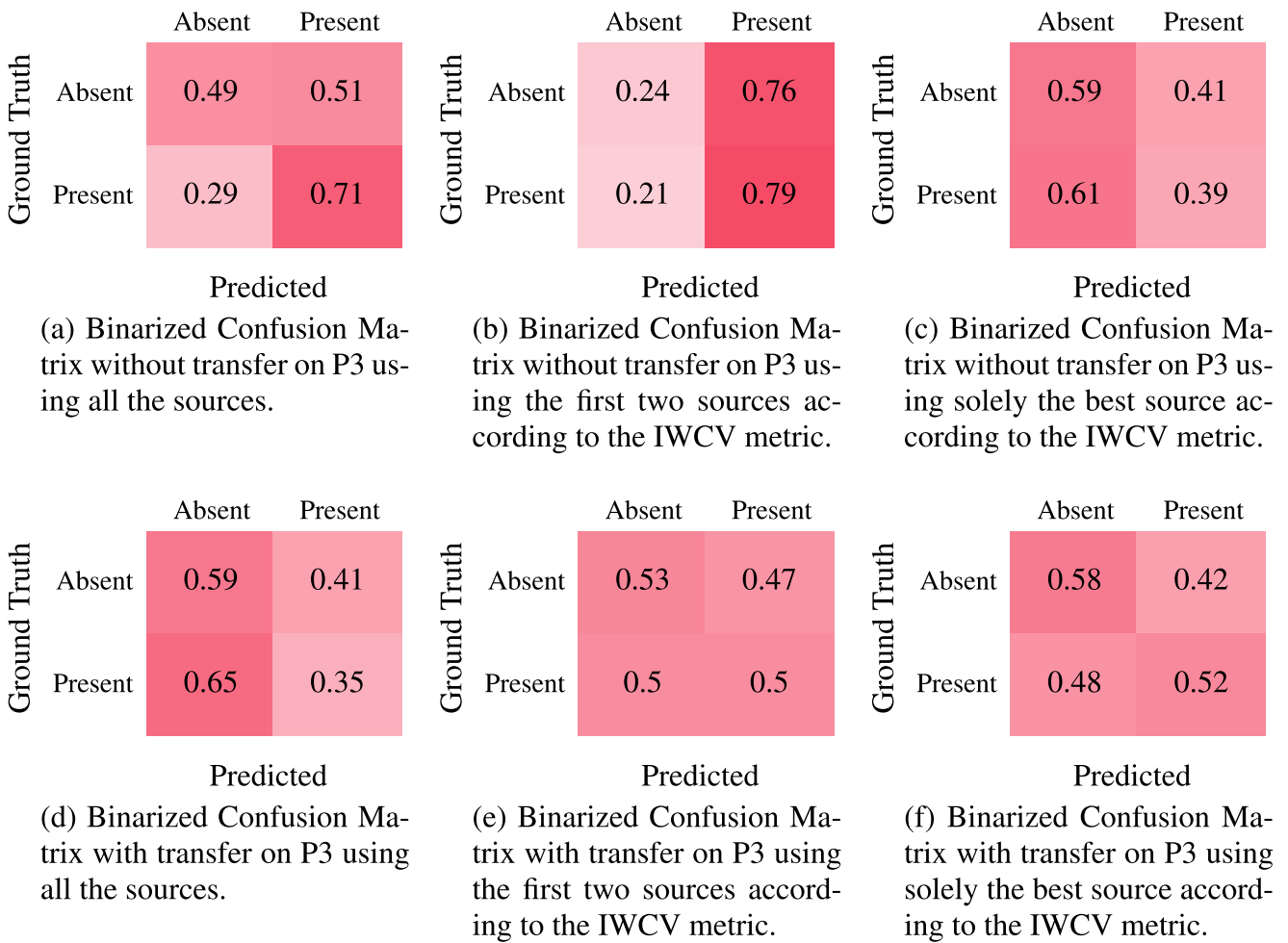


Fig. 8 Binarized Confusion Matrices on P3

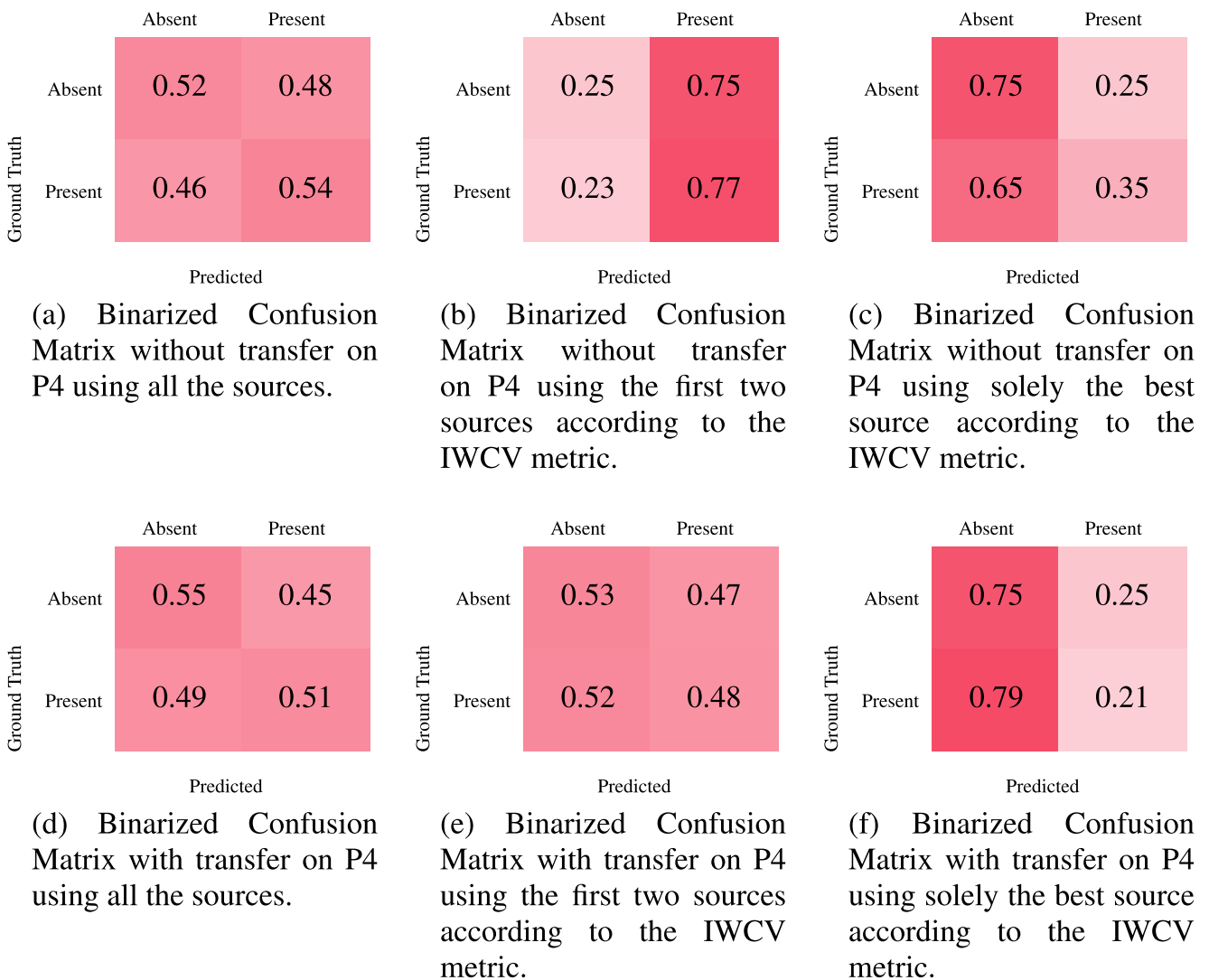


Fig. 9 Binarized Confusion Matrices on P4

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Li X, Zhang D, Liu Z, Li Z, Du C, Dong C (2015) Materials science: Share corrosion data. *Nat News* 527(7579):441
- Nešić S (2007) Key issues related to modelling of internal corrosion of oil and gas pipelines—a review. *Corrosion Sci* 49(12):4308–4338
- De Masi G, Gentile M, Vichi R, Bruschi R, Gabetta G (2017) Multiscale processing of loss of metal: a machine learning approach. In: *Journal of Physics: Conference Series*, vol 869. IOP Publishing, p 012023
- De Waard C, Lotz U (1993) Prediction of CO_2 corrosion of carbon steel. In: *Corrosion-national association of corrosion engineers annual conference*. NACE
- De Masi G, Vichi R, Gentile M, Bruschi R, Gabetta G (2014) A neural network predictive model of pipeline internal corrosion profile. In: *Proceeding of First International Conference on*

- Systems Informatics, Modeling and Simulation, vol 29. IEEE Computer Society, Washington, pp 01–05
6. Liao K, Yao Q, Wu X, Jia W (2012) A numerical corrosion rate prediction method for direct assessment of wet gas gathering pipelines internal corrosion. *Energies* 5(10):3892–3907
 7. Schlumberger Olga dynamic multiphase flow simulator
 8. Canonaco G, Roveri M, Alippi C, Podenzani F, Bennardo A, Conti M, Mancini N (2020) Corrosion prediction in oil and gas pipelines: a machine learning approach. In: 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). IEEE, pp 1–6
 9. Pan SJ, Yang Q (2009) A survey on transfer learning. *IEEE Trans Knowl Data Eng* 22(10):1345–1359
 10. Evgeniou T, Pontil M (2004) Regularized multi-task learning. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 109–117
 11. Duan L, Xu D, Chang S-F (2012) Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 1338–1345
 12. Argyriou A, Evgeniou T, Pontil M (2006) Multi-task feature learning. *Adv Neural Inf Process Syst* 19:41–48
 13. Raina R, Battle A, Lee H, Packer B, Ng AY (2007) Self-taught learning: transfer learning from unlabeled data. In: Proceedings of the 24th international conference on Machine learning, pp 759–766
 14. Pan SJ, Kwok JT, Yang Q et al (2008) Transfer learning via dimensionality reduction. In: AAAI, vol 8, pp 677–682
 15. Long M, Wang J, Ding G, Pan SJ, Philip SY (2013) Adaptation regularization: A general framework for transfer learning. *IEEE Trans Knowl Data Eng* 26(5):1076–1089
 16. Dai W, Yang Q, Xue G-R, Yu Y (2007) Boosting for transfer learning. In: Proceedings of the 24th international conference on Machine learning, pp 193–200
 17. Wu P, Dietterich TG (2004) Improving svm accuracy by training on auxiliary data sources. In: Proceedings of the twenty-first international conference on Machine learning, p 110
 18. Huang J, Gretton A, Borgwardt K, Schölkopf B, Smola A (2006) Correcting sample selection bias by unlabeled data. *Adv Neural Inf Process Syst* 19:601–608
 19. Mihalkova L, Huynh T, Mooney RaJ (2007) Mapping and revising markov logic networks for transfer learning. In: Aaai, vol 7, pp 608–614
 20. Li F, Pan SJ, Jin O, Yang Q, Zhu X (2012) Cross-domain co-extraction of sentiment and topic lexicons. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 410–419
 21. Weiss K, Khoshgoftaar TM, Wang DD (2016) A survey of transfer learning. *J Big data* 3(1):9
 22. Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, Xiong H, He Q (2020) A comprehensive survey on transfer learning. *Proc IEEE* 109(1):43–76
 23. Fishman G (2013) Monte carlo: concepts, algorithms, and applications. Springer Science & Business Media
 24. Shimodaira H (2000) Improving predictive inference under covariate shift by weighting the log-likelihood function. *J Stat Plann Inference* 90(2):227–244
 25. Sugiyama M, Krauledat M, MÄzler K-R (2007) Covariate shift adaptation by importance weighted cross validation. *J Mach Learn Res* 8:985–1005
 26. Sugiyama M, Suzuki T, Kanamori T (2012) Density ratio estimation in machine learning. Cambridge University Press
 27. Scholkopf B, Smola AJ (2018) Learning with kernels: support vector machines, regularization, optimization, and beyond. Adaptive Computation and Machine Learning series
 28. Wu T-F, Lin C-J, Weng RC (2004) Probability estimates for multi-class classification by pairwise coupling. *J Mach Learn Res* 5:975–1005
 29. Platt J et al (1999) Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv Large Margin Class* 10(3):61–74

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Giuseppe Canonaco received the Dr. Eng. degree in Computer Science and Engineering from Politecnico di Milano (Italy) in April 2018. He is a Ph.D. student in Computer Science and Engineering at Politecnico di Milano (Italy). Current research activities deal with Machine Learning-based corrosion prediction, Reinforcement Learning, and Learning in Non-Stationary Environments.



Manuel Roveri received the Dr. Eng. degree in Computer Science Engineering from the Politecnico di Milano (Italy) in June 2003, the MS in Computer Science from the University of Illinois at Chicago (USA) in December 2003 and the Ph.D. degree in Computer Engineering from the Politecnico di Milano (Italy) in May 2007. Currently, he is an Associate Professor at the Department of Electronics and Information of the Politecnico di Milano (Italy). Current research activity addresses Embedded and Edge Artificial Intelligence, Learning in nonstationary/evolving environments and privacy-preserving machine and deep learning. Manuel Roveri is a Senior Member of IEEE and served as Chair and Member in several IEEE Committees. He holds 1 patent and has published about 100 papers in international journals and conference proceedings. He is the recipient of the 2018 IEEE Computational Intelligence Magazine “Outstanding Paper Award” and of the 2016 IEEE Computational Intelligence Society “Outstanding Transactions on Neural Net Systems Paper Award”.



Cesare Alippi is a Professor within the Faculty of Informatics at Università della Svizzera italiana, professor at Politecnico di Milano, Italy, visiting professor at the University of Guangzhou, China and Advisory Professor at the Northwestern Polytechnic in XiAn, China. He was a visiting researcher/professor at UCL (UK), MIT (USA), ESPCI (F), CASIA (RC), A*STAR (SIN), UKobe (JP). Alippi is an IEEE Fellow, Member of the Administrative

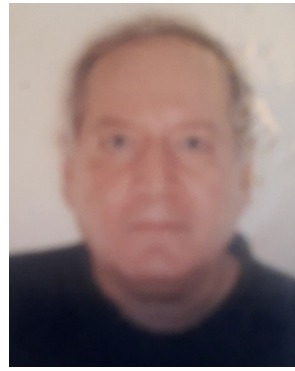
Committee of the IEEE Computational Intelligence Society (CIS), Board of Governors member of the International Neural Network Society; he served the IEEE CIS in several capacities, also covering the roles of Vice-President for education and chairing the Awards committee. He received several awards including the International Neural Networks Society Gabor award. Current research activity addresses adaptation and learning in non-stationary environments, graph learning and Intelligence for embedded, IoT and cyber-physical systems. He holds 8 patents, has published one monograph book, 7 edited books and about 200 papers in international journals and conference proceedings.



Fabrizio Podenzani received the Dr. Eng. degree in Aeronautical Engineering from the Politecnico di Milano (Italy) in 1986. After a brief experience in Agusta Westland (now Leonardo Helicopters), he joined Eni in the 1989 as researcher. Currently he is experimental research technical leader in the Engineering and Modelling Department of New Feedstocks and Energy Vectors Research Center. He has published more than 50

papers in international journals and conference proceedings. He is member of the Working Party in Multiphase Fluid Flow of the European Federation of Chemical Engineer.

Antonio Bennardo received the degree in Nuclear Engineering in February 1990. He joined Eni group since April 1990 and worked as a researcher mainly in the field of computational fluid dynamics. His focus was in the fields of multiphase flow, equipments performances, reactor engineering and multiphase transport in pipelines. He is coauthor of two international patents and about thirty papers in international journals and conference proceeding.



Marco Conti received the Dr. degree in theoretical physics from the Università degli Studi di Milano (Italy) in 1984. He worked in Telettra on semi-conductors and he joined ENI in 1988 in the IT unit as scientific software developer but has developed skills of system and network management in parallel. In 2001 joined the Engineering and Modelling Department of New Feedstocks and Energy Vectors Research Center. He has published a few papers in international journals and conference proceedings.



Nicola Mancini received the degree in Nuclear Engineering in March 1990. Then he joined Eni group and worked as a researcher mainly in the field of computational fluid dynamics. His focus was in the fields of multiphase flow, combustion, reactor engineering and multiphase transport in pipelines. In 2002 he spent three months at Washington University of Saint Louis, working at the CREL (Chemical Reaction Engineering Laboratory) of Prof. Milorad

Dudukovic. He is coauthor of about forty paper in international journals and conference proceeding.