Multi-source transfer learning of time series in cyclical manufacturing

Werner Zellinger, Thomas Grubinger, Michael Zwick, Edwin Lughofer, Holger Schöner, Thomas Natschläger & Susanne Saminger-Platz

Journal of Intelligent Manufacturing

ISSN 0956-5515

J Intell Manuf
DOI 10.1007/s10845-019-01499-4

Springer

Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-019-01499-4

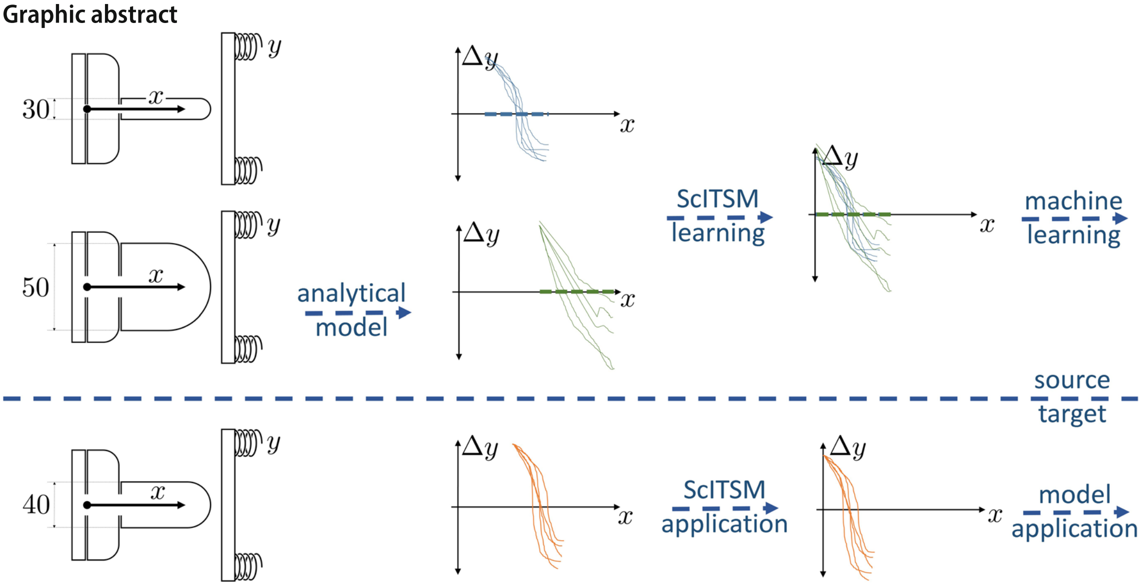
Multi-source transfer learning of time series in cyclical manufacturing

Werner Zellinger[1,2] · Thomas Grubinger[2] · Michael Zwick[2] · Edwin Lughofer[1] · Holger Schöner[2] · Thomas Natschläger[2] · Susanne Saminger-Platz[1]

Received: 23 January 2019 / Accepted: 28 September 2019

Abstract
This paper describes a new transfer learning method for modeling sensor time series following multiple different distributions, e.g. originating from multiple different tool settings. The method aims at removing distribution specific information before the modeling of the individual time series takes place. This is done by mapping the data to a new space such that the representations of different distributions are aligned. Domain knowledge is incorporated by means of corresponding parameters, e.g. physical dimensions of tool settings. Results on a real-world problem of industrial manufacturing show that our method is able to significantly improve the performance of regression models on time series following previously unseen distributions.

Graphic abstract



Keywords Transfer learning · Multi-source transfer learning · Regression · Domain generalization · Domain adaptation

✉ Werner Zellinger
werner.zellinger@jku.at; werner.zellinger@scch.at

Thomas Grubinger
thomasgrubinger@gmail.com

Michael Zwick
michael.zwick@scch.at

Edwin Lughofer
edwin.lughofer@jku.at

Holger Schöner
holger.schoener@siemens.com

Thomas Natschläger
thomas.natschlaeger@scch.at

Susanne Saminger-Platz
susanne.saminger-platz@jku.at

1 Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, Linz, Austria

2 Software Competence Center Hagenberg GmbH, Hagenberg im Mühlkreis, Austria

Published online: 19 October 2019

Introduction

Standard machine learning techniques rely on the assumption that the entire data, both for training and for testing, follows the same distribution. However, this assumption can be violated. In particular, in cyclical manufacturing processes, data is often collected from different operating conditions and environments—called scenarios.

One example is the drilling of steel components (Pena et al. 2005; Ferreiro et al. 2012) where different machine settings can lead to different torque curves during time. A second example is the regression of spectroscopic measurements where different instrumental responses, environmental conditions, or sample matrices can lead to different training and test measurements (Nikzad-Langerodi et al. 2018; Malli et al. 2017). Other examples can be found in the optical inspection of textures or surfaces (Malaca et al. 2016; Stübl et al. 2012; Zăvoianu et al. 2017), where different lightening conditions and texture classes can lead to variations in measurements.

Approaching such heterogeneities in data by standard machine learning techniques requires to model each scenario independently which often causes expensive and time consuming data collection efforts. To overcome this problem, approaches from the field of Transfer Learning (Pan and Yang 2010) have been proposed. Transfer learning aims at extracting knowledge from source scenarios (with large amounts of possibly labeled data) and applies it to the modeling of target scenarios (with little or no available data).

In this paper we address the problem of domain generalization (Muandet et al. 2013), where, assuming enough data from a representative set of (source) scenarios, no data at all is required for the generalization to previously unknown (target) scenarios. We aim at predicting time series from target scenarios arising in cyclical process problems in manufacturing, e.g. torque curves.

We propose a new transfer learning method called *Scenario-Invariant Time Series Mapping (ScITSM)* that leverages available information in multiple similar scenarios and applies it to the prediction of previously unseen scenarios (without available training data).

ScITSM does so by mapping the data in a new space where the scenario-specific data distributions are aligned and such that subsequent joint modeling of the whole transformed data samples is possible. The proposed method is based on the idea of the parameter-based multi-task learning approach presented in Zhang and Yang (2017), where coefficients of neighboring models are either shared or forced to be similar. Our method differs from the approach in Zhang and Yang (2017) by the incorporation of expert knowledge and by its application to time series data. The corrected data from different scenarios is more homogeneous and easier to learn by subsequent machine learning tasks. Furthermore, the learned correction formulas generalize to unseen scenarios. To the best of our knowledge no comparable methods exist that were specifically designed for time series data. The ScITSM method is illustrated in Fig. 1.

The performance of the new algorithm is demonstrated by experiments on a real-world intelligent manufacturing problem. Details of the application must be kept confidential, so it is introduced here in an abstracted way. In particular, a schematic sketch of the application is shown in Fig. 1, the results of the experiments are presented and parts of the collected and preprocessed data are shown. The results indicate that prediction accuracy can be significantly improved by ScITSM.

This paper is organized as follows: Sect. 2 reviews related work, Sect. 3 formulates the problem of domain generalization, Sect. 4 describes the proposed method for Scenario-Invariant Domain Generalization and details our algorithm, Sect. 5 describes our industrial use case, our experiments and results, and, Sect. 6 concludes the work.

Related work

Transfer learning techniques are commonly applied in the areas of computer vision, natural language processing, biology, finance, business management and control application—see e.g. Lu et al. (2015), Grubinger et al. (2016, 2017b), Zellinger et al. (2016, 2017) and references within. Published work in manufacturing applications are relative scare. Successful application in chemistry-oriented manufacturing processes with the usage of chemometric modeling techniques are presented in Nikzad-Langerodi et al. (2018), Malli et al. (2017). Another successful application of transfer learning in intelligent manufacturing for improving product quality was presented in Luis et al. (2010).

The presented method corresponds to the transfer learning subtask of domain generalization (Muandet et al. 2013), which in contrast to other popular transfer learning subtasks like domain adaptation (Zellinger et al. 2017, 2019) does not require any process data measurements of the target scenarios. Many existing domain generalization algorithms can be found in the area of kernel methods (Muandet et al. 2013; Grubinger et al. 2015, 2017a, b; Blanchard et al. 2017; Deshmukh et al. 2017; Gan et al. 2016; Erfani et al. 2016). These algorithms first map the source scenarios in a high dimensional kernel Hilbert space where the different data distributions are aligned and subsequently train a prediction model. Neural network based domain generalization approaches were presented Ghifary et al. (2015), Li et al. (2017a, b). Domain generalization was also combined with SVM (Niu et al. 2015; Xu et al. 2014) and DC-programming (Hoffman et al. 2017).

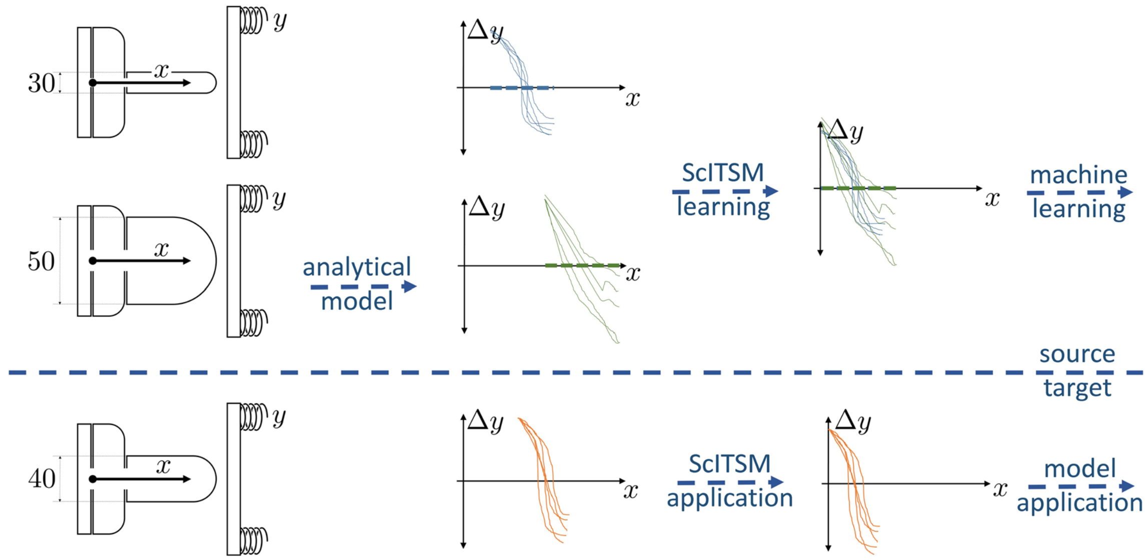


Fig. 1 Schematic sketch of ScITSM for two-source transfer learning with one feature x and target Δy. Left column: Differently parametrized tools acting with feature x on a workpiece causing target feature y. Four basic training steps are performed: (a) Collection of training data from source scenarios (representing tools parametrized by 30 and 50); (b) pre-processing, e.g. analytic modeling, normalization and subsampling; (c) ScITSM for aligning source data distributions (lines in the right column) based on parametric scenario-dependent corrected and smoothed mean curves (dashed lines); (d) training of a single machine learning model based on the aligned data of all source scenarios. The prediction for an unseen target scenario (parametrized by 40) is based on three steps: (a) Collection of target scenario data; (b) application of ScITSM; (c) prediction of Δy using the trained machine learning model

To the best of our knowledge there is no domain generalization method that accounts for multiple source domains and temporal information in time series data.

Formal problem statement

For simplicity, we formulate the problem of multi-source domain generalization for time series of equal length T. Such time series are obtained as results of subsampling procedures as it is the case in our application in Sect. 5.

Following Muandet et al. (2013), Ben-David et al. (2010) and Zellinger et al. (2019), we consider distributions P_1, $\ldots$, P_S and Q over the input space $\mathbb{R}^{N\times T}$ which represent S source scenarios and one target scenario, respectively, where N represents the number of features. In this work, we assume for each of the $S+1$ scenarios a given corresponding parameter vectors $\mathbf{p}_1,\ldots,\mathbf{p}_S,\mathbf{p}_Q\in\mathbb{R}^P$, e.g. corresponding tool dimensions or material properties. Note that the parameter vectors are not the parameters of the distributions $P_1,\ldots,P_n,Q$.

Following Sugiyama and Kawanabe (2012), Ben-David and Urner (2014), we consider an unknown target function $l:\mathbb{R}^{N\times T}\to\mathbb{R}^T$.

Given S source samples $X_1,\ldots,X_S$ drawn from $P_1,\ldots,$ P_S with corresponding target values $Y_1=l(X_1),\ldots,Y_S=l(X_S)$ and parameters $\mathbf{p}_1,\ldots,\mathbf{p}_S$, respectively, the goal of domain generalization is to learn a regression model $f:\mathbb{R}^{N\times T}\to\mathbb{R}^T$ with a small error

$$\mathrm{E}_Q[\|f-l\|]=\int_{\mathbb{R}^{N\times T}}\|f(\mathbf{x})-l(\mathbf{x})\|\;\mathrm{d}\mathbf{x} \tag{1}$$

in the target scenario, where $\|\mathbf{x}\|$ is the Euclidean norm of the vector $\mathbf{x}$. Note that, except for the parameter vector $\mathbf{p}_Q$, no information is given about data in the target scenario.

Scenario-invariant time series mapping

The aim of the proposed ScITSM method is to remove the scenario specific differences in heterogeneous cyclical process manufacturing data such that the transformed data can be jointly modeled by subsequent machine learning procedures. In principle any regression model that accepts time-series data as input can subsequently be employed, e.g. recurrent neural networks or standard machine learning methods based on features contracted from expert knowledge. From our experience, the former usually is the first choice for complex application with very large amounts of available data, while the latter is particularly useful if only a limited amount of data is available.

Theoretical motivation

Intuitively the error in Eq. (1) cannot be small if the target scenario is too different from the source scenarios. However, if the data distributions of the scenarios are similar, this error can be small as shown by the following theorem (obtained as extension of (Ben-David et al. 2010, Theorem 1) to multiple sources and time series).

Theorem 1 *Consider some distributions $P_1, \ldots, P_S$ and Q on the input space $\mathbb{R}^{N\times T}$ and a target function $l : \mathbb{R}^{N\times T} \to [0,1]^T$. Then the following holds for all regression models $f : \mathbb{R}^{N\times T} \to [0,1]^T$:*

$$\mathrm{E}_Q[\|f-l\|] \leq \frac{1}{S}\sum_{i=1}^{S} E_{P_i}[\|f-l\|] + \frac{2\sqrt{T}}{S}\sum_{k=1}^{S} d(P_i, Q) \tag{2}$$

where

$$d(P,Q) = \sup_{B\in\mathcal{B}} |P(B) - Q(B)| \tag{3}$$

is the total variation distance with Borel σ-algebra $\mathcal{B}$.

Proof See "Appendix". □

Theorem 1 shows that the error in the target scenario can be expected to be small if the mean over all errors in the source scenarios is small and the mean distance of the target distribution to the source distributions is small. For simplicity, Theorem 1 assumes a target feature in the unit cube, which can be realized in practice by additional normalization procedures.

Our method tries to minimize the left-hand side of Eq. (2) (target error) by mapping the data in a new space where an approximation of the right-hand side is minimized. The minimization of the second term on the right-hand side is tackled by aligning all source distributions in the new space (they move towards zero in Fig. 1, right column). The minimization of the first term is tackled by subsequent regression.

It is important to note that the alignment of only the source distributions does not minimize the second term on the right-hand side, if the target distribution Q is too different from all the source distributions $P_1, \ldots, P_S$ (Ben-David et al. 2010). As there is no data given from Q in our problem setting (Sect. 3), we cannot identify such cases based on samples. As one possible solution to this problem, we propose to consider only parameter vectors $\mathbf{p}_Q$ which represents physical dimensions of tool settings that are similar to related tool settings represented by $\mathbf{p}_1, \ldots, \mathbf{p}_S$ (see Fig. 2 and compare Fig. 1).

Practical implementation

Consider some source samples $X_1, \ldots, X_S \in \mathbb{R}^{L\times N\times T}$ with target feature vectors $Y_1, \ldots, Y_S \in \mathbb{R}^{L\times T}$ and parameter vectors $\mathbf{p}_1, \ldots, \mathbf{p}_S \in \mathbb{N}^P$ (e.g. parameters 30, 50 in Fig. 1).

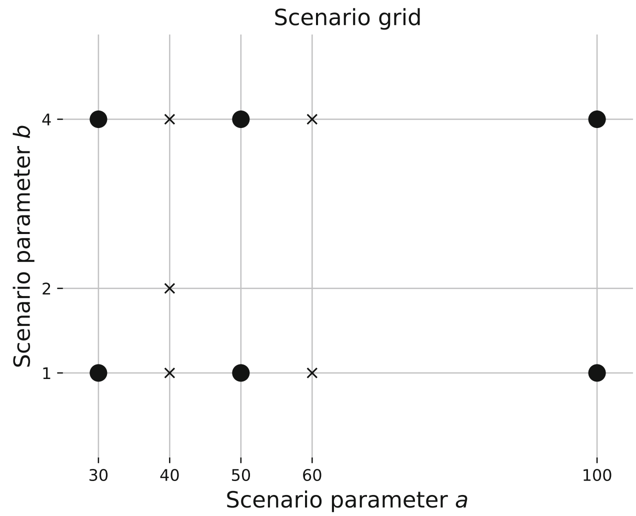


Fig. 2 Use case scenarios with parameters a (horizontal axis) and b (vertical axis). Source scenarios are marked by dots and target scenarios by crosses

For simplicity of the subsequent description, the number of samples L is assumed to be equal for each scenario.

The goal of ScITSM is to compute a mapping

$$\Psi : \mathbb{R}^{N\times T} \times \mathbb{R}^P \to \mathbb{R}^{N\times T} \tag{4}$$

which transforms a time series $\mathbf{x}$ and a scenario parameter vector $\mathbf{p}$ to a new time series $\Psi(\mathbf{x}, \mathbf{p})$ such that the (source) distributions of $\Psi(X_1, \mathbf{p}_1), \ldots, \Psi(X_S, \mathbf{p}_S)$ are similar and such that a subsequently learned regression model $f : \mathbb{R}^T \to \mathbb{R}^T$ performs well on each scenario.

Here, $\Psi(X, \mathbf{p})$ refers to the sample matrix that is obtained by applying $\Psi(\cdot, \mathbf{p})$ to each row of the sample matrix X.

The computation of the function Ψ in ScITSM involves three processing steps: 1. Calculation of a mean curve for each source scenario, 2. Learning of correction functions at equidistant fixed time steps, and, 3. Smooth connection of correction functions.

Step 1: Calculation of mean curves In a first step a smooth curve called *mean curve* is fitted for each source scenario (dashed lines in middle column of Fig. 1).

Therefore, for each of the scenarios samples $X_1, \ldots, X_S$, the mean value for each of the N features and T time steps is computed and a spline curve is fitted subsequently by means of the algorithm proposed in Dierckx (1982). This process results in a matrix $\widehat{X} \in \mathbb{R}^{S\times N\times T}$ storing the mean curves (rows) for each of the S source scenarios.

Step 2: Learning of Equidistant Corrections After the mean curves are computed K equidistant points $t_1, \ldots, t_K$ are fixed and K corresponding *correction functions*

$$\Phi_1, \ldots, \Phi_K : \mathbb{R}^P \to \mathbb{R}^N \tag{5}$$

are learned which map a parameter vector $\mathbf{p}_i$ corresponding to the i-th scenario close to the corresponding points $\widehat{x}_{t_1}, \ldots, \widehat{x}_{t_K}$ of the i-th mean curve $\widehat{\mathbf{x}}_i = (\widehat{x}_1, \ldots, \widehat{x}_T)$, i.e. the i-th row of $\widehat{X}$. This is done under the constraint of similar predictions $\Phi_{t'}(\mathbf{p}_i), \Phi_{t''}(\mathbf{p}_i)$ of nearby time steps t', t'' of two points $\widehat{x}_{t'}, \widehat{x}_{t''}$ on the mean curve.

We apply ideas from the Multi-Task Learning approach proposed in Evgeniou et al. (2004) that aims at similar predictions by means of similar parameters $\theta_1, \ldots, \theta_K$ of the learning functions $\Phi_1, \ldots, \Phi_K$. More precisely, we propose the following objective function:

$$\min_{\Phi_1,\ldots,\Phi_K} \sum_{k=1}^{K} \Bigg(\sum_{i=1}^{S} \left\| \widehat{X}_{i,:,t_k} - \Phi_k(\mathbf{p}_i) \right\| + \alpha \sum_{r=\max(1,k-R)}^{\min(k+R,K)} \frac{\|\theta_k - \theta_r\|^2}{l^{|k-r|-1}} + \beta \, \|\theta_k\|_1 \Bigg) \tag{6}$$

where $X_{i,:,j}$ is the vector of features corresponding to the i-th scenario and the j-th timestep, $\|\cdot\|$ is the Euclidean norm, $\|\cdot\|_1$ is the 1-norm and $\theta_k \in \mathbb{R}^Z$ refers to the parameter vector of Φ_k, e.g. $\Phi_k(\mathbf{p}) = \langle \theta_k, \mathbf{p} \rangle + b$ is a linear model with Euclidean inner product $\langle ., . \rangle$, parameter vector $\theta_k \in \mathbb{R}^P$ and bias $b \in \mathbb{R}$. The first term of Eq. (6) ensures that the prediction of the correction functions applied on the mean curves are not far away from the mean curves itself. The second term of Eq. (6) ensures similar parameter vectors of $2R$ nearby correction functions, where $R \in \mathbb{N}$ and $\alpha, l \in \mathbb{R}$ are hyper-parameters. The last term ensures sparse parameter vectors by means of $L1$-regularization (Andrew and Gao 2007) with hyper-parameter $\beta \in \mathbb{R}$.

Step 3: Smooth Connection To obtain a time series of length T, we aim at a smooth connection of the functions $\Phi_1, \ldots, \Phi_K$ between the points $t_1, \ldots, t_K$. This is done by applying ideas from moving average filtering (Makridakis and Wheelwright 1977). For a new time step $t \leq T$, we denote by

$$\mathcal{R}(t) = \Big\{ \big(\lfloor t \rfloor - R + 1, \lceil t \rceil + R - 1\big), \big(\lfloor t \rfloor - R + 2, \lceil t \rceil + R - 2\big), \ldots, \big(\lfloor t \rfloor, \lceil t \rceil\big) \Big\} \tag{7}$$

a set of pairs constructed from the equidistant timesteps $t_1, \ldots, t_K$ in a nested order, where $\lfloor t \rfloor$ ($\lceil t \rceil$) denotes the largest (smallest) number in $\{t_1, \ldots, t_K\}$ being smaller (larger) than t. The coordinates of the final transformation $\Psi(\mathbf{x}, \mathbf{p}) = (\Psi_1(\mathbf{x}, \mathbf{p}), \ldots, \Psi_T(\mathbf{x}, \mathbf{p}))$ in Eq. (4) are obtained by

$$\Psi_t(\mathbf{x}, \mathbf{p}) = \mathbf{x}_t - \sum_{(i,j)\in\mathcal{R}(t)} \frac{m^{\frac{|\mathcal{R}(t)|-2i+2}{2}} \left(\Phi_i(\mathbf{p}) + (t-i) \frac{\Phi_j(\mathbf{p}) - \Phi_i(\mathbf{p})}{j-i} \right)}{\sum_{(i,j)\in\mathcal{R}(t)} m^{\frac{|\mathcal{R}(t)|-2i+2}{2}}} \tag{8}$$

where $|\mathcal{R}(t)|$ is the cardinality of $\mathcal{R}(t)$ and $m \in (0, 1]$ is the smoothing hyper-parameter. That is, for each vector element $\mathbf{x}_t$ of the time series $\mathbf{x}$, a sum is subtracted which describes a (weighted) average of linear interpolations between the points Φ_i and Φ_j for each time step pair $(i, j) \in \mathcal{R}(t)$. ScITSM is summarized by Algorithm 1.

Algorithm 1: Scenario-Invariant Time Series Mapping (ScITSM)

Input: Samples $X_1, \ldots, X_S \in \mathbb{R}^{L\times N\times T}$ and scenario parameters $\mathbf{p}_1, \ldots, \mathbf{p}_S \in \mathbb{R}^P$

Output: Mapping $\Psi : \mathbb{R}^{N\times T} \times \mathbb{R}^P \to \mathbb{R}^{N\times T}$

Init : Setting of hyper-parameters $\alpha, \beta, l \in \mathbb{R}$, $K, R \in \mathbb{N}$ and $m \in (0, 1]$ and initialization of K correction functions $\Phi_1, \ldots, \Phi_K : \mathbb{R}^P \to \mathbb{R}^N$

Step 1 : Calculation of mean curve tensor $\widehat{X} \in \mathbb{R}^{S\times N\times T}$ as a row-wise concatenation of the means (over rows and columns) of $X_1, \ldots, X_S$.

Step 2 : Computation of correction functions according to Eq. (6).

Step 3 : Computation of transformation Ψ using Eq. (8).

Subsequent regression

Consider a transformation function $\Psi : \mathbb{R}^{N\times T} \times \mathbb{R}^P \to \mathbb{R}^{N\times T}$ as computed by ScITSM, a previously unseen target scenario sample $X_Q = (\mathbf{x}_1, \ldots, \mathbf{x}_L)$ of size L drawn from the unknown target distribution Q over $\mathbb{R}^{N\times T}$ and a corresponding parameter vector $\mathbf{p}_Q \in \mathbb{N}^P$ (e.g. parameter 40 in Fig. 1). As motivated in Sect. 4.1, the distribution of the transformed sample $\Psi(X_Q, \mathbf{p}_Q)$ is assumed to be similar to the distributions of the samples $\Psi(X_1, \mathbf{p}_1), \ldots, \Psi(X_S, \mathbf{p}_S)$ which is induced by the selection of an appropriate corresponding parameter space (see e.g. Figs. 1 and 2).

Subsequently to ScITSM, a regression function

$$f : \mathbb{R}^{N\times T} \to \mathbb{R}^T \tag{9}$$

is trained using the concatenated input sample $(\Psi(X_1, \mathbf{p}_1); \ldots; \Psi(X_S, \mathbf{p}_S))$ and its corresponding target features $(Y_1; \ldots; Y_S)$. Finally, the target features of X_Q can be computed by $f(\Psi(X_Q, \mathbf{p}_Q))$.

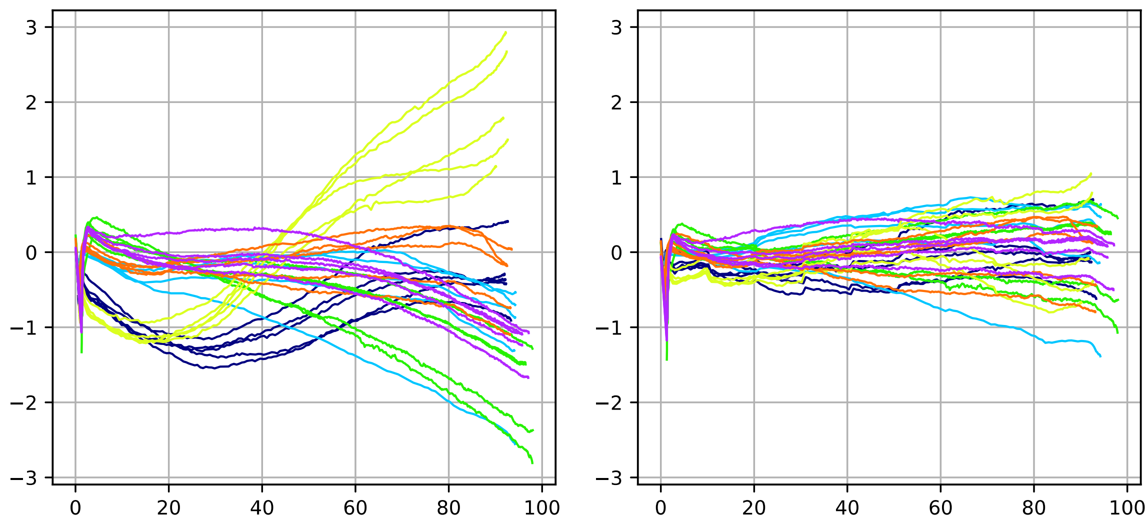


Fig. 3 Some selected pre-processed time series of source scenarios (different colors) before (left) and after (right) the application of ScITSM (Color figure online)

Theorem 1 shows that the empirical error

$$\frac{1}{N}\sum_{i=1}^{N}\left\| f(\Psi(\mathbf{x}_i,\mathbf{p}_Q)) - l(\mathbf{x}_i)\right\| \tag{10}$$

of the function $f \circ \Psi$ on the target sample can be expected to be small, if the sample size is large enough (i.e. the empirical error approximates well the error in Eq. 1) and the model f performs well on the concatenated source data.

Use case

Intelligent manufacturing extends control systems with machine learning models trained from gathered data, e.g. Virtual Sensors (Wang and Nace 2009). We integrated our approach described in Sect. 4 into the data-flow of a machine learning pipeline used to implement a Virtual Sensor in an Intelligent Manufacturing setting similar to the one described in Fig. 1.

Dataset

Our use case consists of 11 scenarios based on physical tool settings with parameters describing physical tool dimensions as illustrated in Fig. 2. For each scenario, we collected around 50 time series. We applied some application-specific normalization and transformation steps to each time series including its subtraction from a finite element simulation of the mechanical tool process. Some representative resulting time series from the source scenarios are illustrated in Fig. 3 on the left. For our experiments we choose 6 (out of 11) scenarios as source scenarios and 5 scenarios as target scenarios. The target scenarios are chosen such that its parametrization is well captured by the parametrization of the source scenarios (see Fig. 2).

Validation procedure

To estimate the performance of the proposed ScITSM on previously unseen scenarios, we evaluate different regression models based on an unsupervised transductive training protocol (Ganin et al. 2016; Gong et al. 2013; Chopra et al. 2013; Long et al. 2017) combined with cross-validation on source scenarios.

In a first step, we select appropriate hyper-parameters in a semi-automatic way. That is, the parameters are fixed by a method expert based only on the unsupervised data from the source scenarios without considering any labels, i.e. output values, or target samples. The decision is based on visual quantification of the distribution alignment in the representation space. As a result, the hyper-parameters are the same for all subsequently trained regression models. The result of some representative time series is illustrated in Fig. 3.

For evaluating the performance of regression models trained subsequently to ScITSM we use 10-fold cross-validation (Varma and Simon 2006). That is, in each of 10 steps, 90% of the data (90% of each source scenario) are chosen as training data and 10% as validation data.

Since no data of the target scenarios is used for training, the models are evaluated on the whole data of the target scenarios in each fold.

Using this protocol, 10 different root-mean squared errors for each model and each scenario are computed, properly aggregated and (together with its standard deviation) reported in Table 1.

To show the advantage of using more than one source scenario, we additionally optimize each regression model

Table 1 Root mean squared error (and standard deviation) of regression models evaluated using tenfold cross-validation as described in Sect. 5.2

Scenario	Without ScITSM	With ScITSM	Perc.	Without ScITSM	With ScITSM	Perc.
	Bayesian ridge			Random forest		
(1, 30)	0.443 (0.082)	0.239 (0.056)	*53.93*	0.259 (0.109)	0.262 (0.082)	101.13
(1, 50)	0.645 (0.070)	0.359 (0.103)	*55.69*	0.322 (0.140)	0.311 (0.111)	*96.62*
(1, 100)	0.431 (0.140)	0.299 (0.070)	*69.34*	0.308 (0.090)	0.267 (0.064)	*86.48*
(4, 30)	0.690 (0.117)	0.334 (0.077)	*48.47*	0.346 (0.095)	0.372 (0.064)	107.31
(4, 50)	0.431 (0.052)	0.243 (0.090)	*56.44*	0.317 (0.098)	0.238 (0.051)	*75.11*
(4, 100)	0.488 (0.105)	0.235 (0.064)	*48.05*	0.197 (0.077)	0.234 (0.101)	118.87
Average	0.521 (0.094)	0.285 (0.077)	*55.32*	0.292 (0.102)	0.281 (0.079)	*97.59*
(1, 40)	0.523 (0.078)	0.403 (0.125)	*77.12*	0.707 (0.243)	0.418 (0.163)	*59.12*
(1, 60)	0.709 (0.058)	0.394 (0.087)	*55.54*	0.461 (0.148)	0.381 (0.099)	*82.72*
(2, 40)	0.576 (0.092)	0.426 (0.117)	*73.90*	0.949 (0.236)	0.440 (0.108)	*46.34*
(4, 40)	0.426 (0.031)	**0.342 (0.076)**	*80.30*	1.062 (0.238)	0.399 (0.114)	*37.57*
(4, 60)	0.519 (0.110)	0.371 (0.142)	*71.58*	**0.291 (0.060)**	0.395 (0.165)	135.76
Average	0.551 (0.074)	0.387 (0.109)	*71.69*	0.694 (0.185)	0.407 (0.130)	*72.30*
	SVR (sigmoid)			SVR (RBF)		
(1, 30)	0.586 (0.114)	0.253 (0.081)	*43.17*	0.243 (0.072)	**0.238 (0.068)**	*97.64*
(1, 50)	0.519 (0.221)	0.364 (0.170)	*70.15*	0.229 (0.092)	**0.226 (0.078)**	*98.46*
(1, 100)	0.694 (0.202)	0.379 (0.159)	*54.63*	0.249 (0.064)	**0.242 (0.070)**	*97.26*
(4, 30)	1.697 (0.341)	0.407 (0.067)	*23.97*	0.342 (0.122)	**0.294 (0.098)**	*85.95*
(4, 50)	0.363 (0.154)	0.325 (0.141)	*89.66*	0.201 (0.060)	**0.192 (0.042)**	*95.71*
(4, 100)	0.682 (0.199)	0.341 (0.090)	*49.93*	0.186 (0.059)	**0.166 (0.032)**	*89.00*
Average	0.757 (0.205)	0.345 (0.118)	*55.25*	0.242 (0.078)	**0.226 (0.065)**	*93.28*
(1, 40)	0.491 (0.142)	0.483 (0.134)	*98.34*	0.445 (0.151)	**0.387 (0.129)**	*87.13*
(1, 60)	0.637 (0.208)	0.450 (0.134)	*70.70*	0.337 (0.079)	**0.321 (0.064)**	*95.24*
(2, 40)	0.518 (0.085)	0.570 (0.158)	109.95	**0.314 (0.055)**	0.385 (0.096)	122.72
(4, 40)	0.684 (0.189)	0.452 (0.153)	*66.08*	0.382 (0.156)	0.378 (0.156)	*98.66*
(4, 60)	0.507 (0.202)	0.487 (0.196)	*96.08*	0.334 (0.056)	0.339 (0.134)	101.45
Average	0.567 (0.165)	0.488 (0.155)	*88.23*	**0.362 (0.099)**	0.363 (0.116)	101.04

Best values of scenarios are shown in boldface, improvements of ScITSM are shown by italic numbers

using the training data of only a single source scenario (see Table 2).

We compare the following regression models and we use the following parameter sets for selection:

- Bayesian Ridge Regression (MacKay 1992): The four gamma priors are searched in the set $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$ and the iterative algorithm is stopped when a selected error in the set $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ is reached.
- Random Forest (Breiman 2001): We used 100 estimators, the maximum depth is searched in the set $\{1, 2, 4, 8, \ldots, \infty\}$ where ∞ refers to a pure expansion of the leaves and the minimum number of splits is selected in the set $\{2, 4, 8, \ldots, 1024\}$.
- Support Vector Regression (Smola and Schölkopf 2004) (SVR) with sigmoid kernel: The epsilon parameter is selected from the set $\{10^{-1}, 10^{-2}, 10^{-3}\}$, the parameter C is selected in $\{10^{-5}, 5 \cdot 10^{-4}, 10^{-4}, 5 \cdot 10^{-3}, 10^{-3}\}$ and the algorithm is stopped when a selected error in the set $\{10^{-3}, 10^{-5}\}$ is reached.
- Support Vector Regression with RBF kernel: The epsilon parameter is selected from the set $\{10^{-1}, 10^{-2}, 10^{-3}\}$, the parameter C is selected in $\{10, 25, 30\}$, the bandwidth parameter is selected in the set $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ and the algorithm is stopped when a selected error in the set $\{10^{-3}, 10^{-2}\}$ is reached.

Results

Figure 3 illustrates some selected time series pre-processed by ScITSM. It can be seen, that the diversity caused by different source scenarios is reduced resulting in more homogeneous time series for subsequent regression.

Table 2 Root mean squared error (and standard deviation) of regression models trained and evaluated on a single source scenario, i.e. one model per scenario, using tenfold cross-validation as described in Sect. 5.2

Scenario	Without ScITSM	With ScITSM	Perc.	Without ScITSM	With ScITSM	Perc.
	Bayesian ridge			Random forest		
(1,30)	0.215 (0.069)	0.210 (0.065)	*97.66*	0.255 (0.079)	0.261 (0.078)	102.15
(1,50)	0.202 (0.047)	0.202 (0.048)	100.00	0.370 (0.172)	0.352 (0.151)	*95.05*
(1,100)	0.342 (0.112)	0.341 (0.109)	*99.67*	0.325 (0.100)	0.330 (0.127)	101.55
(4,30)	0.275 (0.072)	0.275 (0.074)	100.09	0.351 (0.090)	0.334 (0.094)	*95.00*
(4,50)	0.217 (0.069)	0.217 (0.070)	100.00	0.301 (0.091)	0.292 (0.081)	*96.84*
(4,100)	0.197 (0.057)	0.196 (0.058)	*99.42*	0.240 (0.058)	0.269 (0.095)	111.70
	SVR (sigmoid)			SVR (RBF)		
(1,30)	0.404 (0.096)	0.273 (0.099)	*67.54*	0.390 (0.157)	0.380 (0.161)	*97.42*
(1,50)	0.486 (0.223)	0.394 (0.222)	*81.01*	0.364 (0.173)	0.357 (0.159)	*98.12*
(1,100)	0.656 (0.229)	0.405 (0.167)	*61.72*	0.360 (0.201)	0.369 (0.194)	102.24
(4,30)	1.130 (0.149)	0.440 (0.071)	*38.97*	0.502 (0.298)	0.438 (0.244)	*87.17*
(4,50)	0.382 (0.176)	0.354 (0.174)	*92.76*	0.323 (0.108)	0.322 (0.110)	*99.67*
(4,100)	0.580 (0.181)	0.364 (0.094)	*62.80*	0.215 (0.080)	0.234 (0.102)	108.64

Improvements of ScITSM are shown by italic numbers

Table 1 shows the results of applying ScITSM to multiple source scenarios. The application of ScISTM improves all regression models in average root mean squared error except the support vector regression model based on RBF kernel.

The scenario (2, 40) is the only scenario where the application of ScITSM reduces the performance of support vector regression models by a large margin. From Fig. 2 it can be seen that both tool dimensions 2 and 40 are not considered in the source scenarios. We conclude that at least one dimension should be considered in the source scenarios in our use case, otherwise the scenario distributions are too different. This well known phenomenon is often called *negative transfer* (Pan et al. 2010).

It is interesting to observe that the random forest models 'overfit' the source scenarios. This can be seen by a low average root mean squared error on the source scenarios compared to the target scenarios. Consequently, it is hard for ScITSM to improve the performance on the source scenarios (average error decreased to 97.59% of that of the raw models) where the target scenarios errors are improved by a large margin. The target scenario improvement is without considering scenario (4, 60) where the random forest model performed best over all models. This improvement is not unexpected, as the 'overfitting' of source scenarios can imply performance improvements in some very similar target scenarios. However, our goal is an improvement in many scenarios, not in single ones.

In general ScITSM improves the results of regression models in 9 out of 11 scenarios, where the remaining two results have explainable reasons of negative transfer and overfitting.

In principle it is possible that a high root mean squared error of the models without ScITSM is caused by mixing data from different scenarios, i.e. negative transfer happens. To exclude this possibility, we trained one model for each scenario and computed the root mean squared error for all other scenarios.

In a first step, we observed that no model is able to generalize to scenarios other than the single training one. The resulting root mean squared errors of the single scenario trained models are excessively high and give no further information that can be reported in this work. One possible reason is that the scenarios are too different. For example, consider a model trained on the yellow time series in Fig. 3 on the left and tested on data on the green time series. This experiment underpins that generalization is not possible for models trained only on single scenarios (standard regression case) and that the considered problem of domain generalization is important in our use case.

It is interesting to observe that even models trained on single scenarios (standard regression case) can be improved by considering data from different scenarios. To see this, consider Table 2. Each column denoted by 'without ScITSM' shows the performance of different models trained on data from a single scenario only (shown by the row). This is in contrast to Table 1 where each column shows errors of the same model on different scenarios. Applying ScITSM to data from other scenarios, almost always improves the performance of (standard) regression models. This is interesting as one may expect that models trained on data from a specific scenario cannot be improved by data from different scenarios. However, this positive effect of transfer learning can happen when

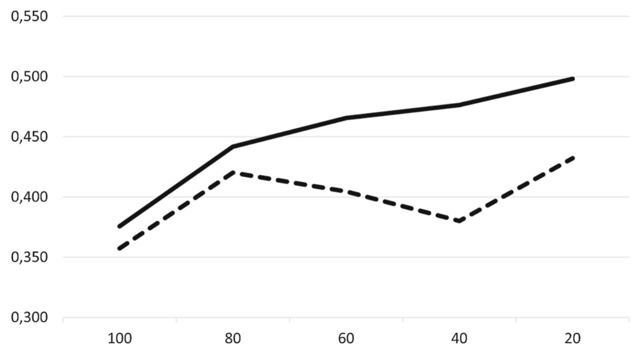


Fig. 4 Performance dependency on sample size of support vector regression with RBF kernel without applying ScITSM (solid) and with the proposed ScITSM (dashed). Horizontal axis: Percentage of training data; Vertical axis: Average root mean squared error over all unseen target scenarios except negative transfer scenario (2, 40)

a high number of scenarios is considered with a comparably low amount of samples.

Another interesting question is about the effect of ScITSM when the amount of source scenario samples decreases. Therefore, we consider the average root mean squared errors over all target scenarios of the best regression models, i.e. SVR with RBF kernel, for a varying number of source samples. The result is shown in Fig. 4. It can be seen that the positive effect of ScITSM gets even stronger when the sample size of all scenarios decreases by a certain percentage value.

Our procedure of choosing appropriate parameters for ScITSM requires expert knowledge about our method. In our use case, long-term knowledge from several years resulted in a well-performing default setting. It is interesting to observe that this default setting gives a high performance independently of the data size (see Fig. 4). It is important to note that the selection of appropriate parameters is sophisticated in the considered problem of domain generalization, as no data of the target scenarios is given. No classical cross-validation procedures can be used which would suffer from an unbounded bias in the generalization error estimate (Zhong et al. 2010). Finding appropriate parameters for transfer learning is an active research area (You et al. 2019). Most methods rely on a small set of data from the target scenarios (Long et al. 2012; Ganin et al. 2016) or fix their parameters (Zellinger et al. 2017) to some default values. Unfortunately, both variants cannot be used in our industrial use case. Note, by using this method, the resulting performance of the regression models in the source scenarios cannot be directly interpreted as estimating the generalization error. However, in this work, we are more interested in the generalization error of the unseen target scenarios, which are not effected.

We finally conclude that our method successfully enables the improvement of the performance of regression models in previously unseen scenarios by using information from multiple similar source scenarios. The result is obtained by a single regression model, which is conceptually and computationally simpler than the application of multiple single models for separate scenarios.

Conclusion and future work

A multi-source transfer learning method for time series data is proposed. The method transforms the data in a new space such that the distributions of samples produced by multiple different tool settings are aligned. Domain knowledge is incorporated by means of corresponding tool dimensions. In a real world application of industrial manufacturing, the proposed methods significantly reduce the prediction error on data originating from already seen tool settings. The biggest benefit of the proposed method is that it can be applied to unseen data from new unseen tool settings without the need of time and cost intensive collection of training data using these settings.

Unfortunately, parameter selection becomes an important issue without data from unseen tool settings. Without such data, it is also hard to identify wrong expert knowledge used in our work to select appropriate future settings.

However, small amounts of (possibly unlabeled) data from new tool settings could be used to improve the parameter selection process in the future. These small amounts of data could also be used to overcome the phenomenon of negative transfer by strengthening the similarity assessment of data distributions from different tool settings.

Acknowledgements Open access funding provided by Johannes Kepler University Linz. This work was partially funded by SCCH within the Austrian COMET programme. We thank Ciprian Zavoianu for helpful discussions. The fourth author acknowledges the support by the LCM-K2 Center within the framework of the Austrian COMET-K2 program.

Appendix: Proof of Theorem 1

Proof Let $p_1, \ldots, p_S$ and q be the probability density functions of $P_1, \ldots, P_S$ and Q, respectively. Then the following holds:

$$\mathrm{E}_Q\left[\|f-l\|\right] = \int \|f-l\|\, q$$
$$+ \frac{1}{S}\sum_{i=1}^{S}\int \|f-l\|\, p_i - \frac{1}{S}\sum_{i=1}^{S}\int \|f-l\|\, p_i$$

$$
\begin{aligned}
&= \frac{1}{S}\sum_{i=1}^{S}\int \|f-l\|\, p_i + \int \|f-l\| \left(q - \frac{1}{S}\sum_{i=1}^{S} p_i\right) \\
&\le \frac{1}{S}\sum_{i=1}^{S}\int \|f-l\|\, p_i + \int \|f-l\| \left|q - \frac{1}{S}\sum_{i=1}^{S} p_i\right| \\
&= \frac{1}{S}\sum_{i=1}^{S}\int \|f-l\|\, p_i \\
&\quad + \int \|f-l\| \left|\frac{1}{S}\sum_{i=1}^{S} q - \frac{1}{S}\sum_{i=1}^{S} p_i\right| \\
&\le \frac{1}{S}\sum_{i=1}^{S}\int \|f-l\|\, p_i + \int \|f-l\| \frac{1}{S}\sum_{i=1}^{S} |q-p_i| \\
&\le \frac{1}{S}\sum_{i=1}^{S} \mathrm{E}_{P_i}\left[\|f-l\|\right] + \sup_{\mathbf{x}\in\mathbb{R}^{N\times T}} \|f(\mathbf{x}) - l(\mathbf{x})\| \\
&\quad \times \int \frac{1}{S}\sum_{i=1}^{S} |q-p_i| \le \frac{1}{S}\sum_{i=1}^{S} \mathrm{E}_{P_i}\left[\|f-l\|\right] \\
&\quad + \sup_{\mathbf{y},\mathbf{y}'\in[0,1]^T} \left\|\mathbf{y}-\mathbf{y}'\right\| \frac{1}{S}\sum_{i=1}^{S}\int |q-p_i| \\
&= \frac{1}{S}\sum_{i=1}^{S} \mathrm{E}_{P_i}\left[\|f-l\|\right] + \sup_{\mathbf{x}\in[-1,1]^T} \|\mathbf{x}\| \frac{1}{S}\sum_{i=1}^{S}\int |q-p_i| \\
&= \frac{1}{S}\sum_{i=1}^{S} \mathrm{E}_{P_i}\left[\|f-l\|\right] + \frac{\sqrt{T}}{S}\sum_{i=1}^{S}\int |q-p_i| \\
&= \frac{1}{S}\sum_{i=1}^{S} \mathrm{E}_{P_i}\left[\|f-l\|\right] + \frac{2\sqrt{T}}{S}\sum_{i=1}^{S} d(P_i, Q)
\end{aligned}
$$

where the last equality follows from the application of Lemma 2.1 in Tsybakov (2008). □

References

Andrew, G., & Gao, J. (2007). Scalable training of L1-regularized log-linear models. In *Proceedings of the international conference on machine learning* (pp. 33–40).

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, *79*(1–2), 151–175.

Ben-David, S., & Urner, R. (2014). Domain adaptation-can quantity compensate for quality? *Annals of Mathematics and Artificial Intelligence*, *70*(3), 185–202.

Blanchard, G., Deshmukh, A. A., Dogan, U., Lee, G., & Scott, C. (2017). Domain generalization by marginal transfer learning. arXiv preprint arXiv:1711.07910.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Chopra, S., Balakrishnan, S., & Gopalan, R. (2013). Dlid: Deep learning for domain adaptation by interpolating between domains. In *International conference on machine learning workshop on challenges in representation learning*.

Deshmukh, A. A., Sharma, S., Cutler, J. W., & Scott, C. (2017). Multi-class domain generalization. In *NIPS workshop on limited labeled data*.

Dierckx, P. (1982). A fast algorithm for smoothing data on a rectangular grid while using spline functions. *SIAM Journal on Numerical Analysis*, *19*(6), 1286–1304.

Erfani, S., Baktashmotlagh, M., Moshtaghi, M., Nguyen, V., Leckie, C., Bailey, J., & Kotagiri, R. (2016). Robust domain generalisation by enforcing distribution invariance. In *Proceedings of the international joint conference on artificial intelligence* (pp. 1455–1461). AAAI Press/International Joint Conferences on Artificial Intelligence.

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). ACM.

Ferreiro, S., Sierra, B., Irigoien, I., & Gorritxategi, E. (2012). A Bayesian network for burr detection in the drilling process. *Journal of Intelligent Manufacturing*, *23*(5), 1463–1475.

Gan, C., Yang, T., & Gong, B. (2016). Learning attributes equals multi-source domain generalization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 87–97).

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., et al. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, *17*, 1–35.

Ghifary, M., Bastiaan Kleijn, W., Zhang, M., & Balduzzi, D. (2015). Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision* (pp. 2551–2559).

Gong, B., Grauman, K., & Sha, F. (2013). Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of the international conference on machine learning* (pp. 222–230).

Grubinger, T., Birlutiu, A., Schöner, H., Natschläger, T., & Heskes, T. (2015). Domain generalization based on transfer component analysis. In I. Rojas, G. Joya, & A. Catala (Eds.), *Advances in computational intelligence* (pp. 325–334). Berlin: Springer.

Grubinger, T., Birlutiu, A., Schöner, H., Natschläger, T., & Heskes, T. (2017a). Multi-domain transfer component analysis for domain generalization. *Neural Processing Letters*, *46*, 1–11.

Grubinger, T., Chasparis, G. C., & Natschläger, T. (2016). Online transfer learning for climate control in residential buildings. In *Proceedings of the annual European control conference (ECC 2016)* (pp. 1183–1188).

Grubinger, T., Chasparis, G. C., & Natschläger, T. (2017b). Generalized online transfer learning for climate control in residential buildings. *Energy and Buildings*, *139*, 63–71.

Hoffman, J., Mohri, M., & Zhang, N. (2017). Multiple-source adaptation for regression problems. arXiv preprint arXiv:1711.05037.

Li, D., Yang, Y., Song, Y. Z., & Hospedales, T. M. (2017a). Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision* (pp. 5543–5551).

Li, D., Yang, Y., Song, Y. Z., & Hospedales, T. M. (2017b). Learning to generalize: Meta-learning for domain generalization. arXiv preprint arXiv:1710.03463

Long, M., Wang, J., Ding, G., Shen, D., & Yang, Q. (2012). Transfer learning with graph co-regularization. In *Conference on artificial intelligence* (pp. 1805–1818). AAAI.

Long, M., Wang, J., & Jordan, M. I. (2017). Deep transfer learning with joint adaptation networks. In *Proceedings of the international conference on machine learning*.

Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., & Zhang, G. (2015). Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems*, *80*, 14–23.

Luis, R., Sucar, L. E., & Morales, E. F. (2010). Inductive transfer for learning bayesian networks. *Machine Learning*, *79*(1–2), 227–255.

MacKay, D. J. (1992). Bayesian interpolation. *Neural Computation*, *4*(3), 415–447.

Makridakis, S., & Wheelwright, S. C. (1977). Adaptive filtering: An integrated autoregressive/moving average filter for time series forecasting. *Journal of the Operational Research Society*, *28*(2), 425–437.

Malaca, P., Rocha, L. F., Gomes, D., Silva, J., & Veiga, G. (2016). Online inspection system based on machine learning techniques: Real case study of fabric textures classification for the automotive industry. *Journal of Intelligent Manufacturing*, *30*, 1–11.

Malli, B., Birlutiu, A., & Natschläger, T. (2017). Standard-free calibration transfer: An evaluation of different techniques. *Chemometrics and Intelligent Laboratory Systems*, *161*, 49–60.

Muandet, K., Balduzzi, D., & Schölkopf, B. (2013). Domain generalization via invariant feature representation. In *Proceedings of the 30th international conference on machine learning* (pp. 10–18).

Nikzad-Langerodi, R., Zellinger, W., Lughofer, E., & Saminger-Platz, S. (2018). Domain-invariant partial least squares regression. *Analytical Chemistry*, *90*, 6693.

Niu, L., Li, W., & Xu, D. (2015). Multi-view domain generalization for visual recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 4193–4201).

Pan, S., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Pan, S., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Pena, B., Aramendi, G., Rivero, A., & de Lacalle, L. N. L. (2005). Monitoring of drilling for burr detection using spindle torque. *International Journal of Machine Tools and Manufacture*, *45*(14), 1614–1621.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222.

Stübl, G., Bouchot, J. L., Haslinger, P., & Moser, B. (2012). Discrepancy norm as fitness function for defect detection on regularly textured surfaces. In: Joint DAGM (German Association for Pattern Recognition) and OAGM symposium (pp. 428–437). Springer.

Sugiyama, M., & Kawanabe, M. (2012). *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. Cambridge: MIT Press.

Tsybakov, A. B. (2008). *Introduction to nonparametric estimation* (1st ed.). Berlin: Springer.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, *7*(1), 91.

Wang, L., & Nace, A. (2009). A sensor-driven approach to web-based machining. *Journal of Intelligent Manufacturing*, *20*(1), 1–14.

Xu, Z., Li, W., Niu, L., & Xu, D. (2014). Exploiting low-rank structure from latent domains for domain generalization. In *Proceedings of the European conference on computer vision* (pp. 628–643).

You, K., Wang, X., Long, M., & Jordan, M. (2019). Towards accurate model selection in deep unsupervised domain adaptation. In *Proceedings of the international conference on machine learning* (pp. 7124–7133).

Zăvoianu, A. C., Lughofer, E., Pollak, R., Meyer-Heye, P., Eitzinger, C., & Radauer, T. (2017). Multi-objective knowledge-based strategy for process parameter optimization in micro-fluidic chip production. In *IEEE symposium series on computational intelligence* (pp. 1–8). IEEE.

Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T., & Saminger-Platz, S. (2017). Central moment discrepancy (CMD) for domain-invariant representation learning. In *International conference on learning representations*. https://openreview.net/pdf?id=SkB-_mcel.

Zellinger, W., Moser, B., Chouikhi, A., Seitner, F., Nezveda, M., & Gelautz, M. (2016). Linear optimization approach for depth range adaption of stereoscopic videos. Stereoscopic displays and applications XXVII, IS&T Electronic Imaging.

Zellinger, W., Moser, B. A., Grubinger, T., Lughofer, E., Natschläger, T., & Saminger-Platz, S. (2019). Robust unsupervised domain adaptation for neural networks via moment alignment. *Information Sciences*, *483*, 174–191.

Zhang, Y., & Yang, Q. (2017). A survey on multi-task learning. CoRR http://arxiv.org/abs/1707.08114.

Zhong, E., Fan, W., Yang, Q., Verscheure, O., & Ren, J. (2010). Cross validation framework to choose amongst models and datasets for transfer learning. In *Proceedings of the Joint European conference on machine learning and knowledge discovery in databases* (pp. 547–562). Springer.