Process Monitoring using Domain-Adversarial Probabilistic Principal Component Analysis: A Transfer Learning Framework

Atefeh Daemi, Bhushan Gopaluni, Biao Huang, *Fellow, IEEE*

***Abstract*—Probabilistic principal component analysis (PPCA) is a feature extraction method that has been widely used in the field of process monitoring. However, PPCA assumes that training and testing data are drawn from the same input feature space with the same distributions. This assumption is not valid for complex processes that exhibit multiple operating modes and generate data with different distributions. We propose a novel transfer learning approach to monitoring processes with data from multiple distributions. To this end, we introduce a novel extension of probabilistic principal component analysis, which is we refer to as the Domain Adversarial Probabilistic Principal Component Analysis (DAPPCA). DAPPCA algorithm automatically learns feature representations that are relevant across different operational modes. The algorithm extracts the most informative shared fault features and improves the accuracy of the fault detection model in a new operating mode using the knowledge transferred from previously known modes. The parameters of DAPPCA are estimated using a variational inference approach, and the monitoring statistics are calculated using the proposed model. We demonstrate the efficacy and real-time applicability of the proposed method with simulated and industrial examples.**

***Index Terms*—Transfer Learning, Probabilistic Principal Component Analysis, Domain-Adversarial, Variational Inference, Process Monitoring**

I. Introduction

MULTIVARIATE statistical process monitoring has received significant attention in the process control community over the recent decades due to a large number of variables measured and collected. The main objective of this research area is to monitor the performance of processes to ensure their long-term operational reliability and safety. Multivariate statistical process monitoring methods form the data-driven models from normal operating data using principal component analysis (PCA), partial least squares (PLS), or other variants of classical latent variable models [1]–[4]. The measurements are decomposed into a principal component subspace (PCS) and a residual subspace (RS) within these methods. Two commonly used indices for detecting abnormal conditions, namely the Hotelling's T^2 statistic and the squared prediction error (SPE) statistic, are based on the squared Mahalanobis distance of the PCS and the RS, respectively. Fault detection algorithms that detect the process deviations from the normal operating conditions are used to extract valuable indices from data to indicate the process operating status. The occurrence of a fault would lead to an increase in T^2 and SPE to the degree that exceeds the control limits. These control limits are defined such that almost all of the data corresponding to the normal operating condition are within these limits.

On the other hand, the probabilistic counterparts of the classical latent variable models have also been studied widely in the area of process monitoring [5]–[11]. These probabilistic variants have been introduced to address the challenges associated with missing and multi-modality of process data by assuming different distributions for the data.

However, the methods mentioned above for process monitoring perform poorly when the faulty data are limited, as is often the case with real industrial data. As an example, chemical processes that exhibit multiple operating modes often have a limited amount of data corresponding to each fault. In general, failure events are rare, and therefore the corresponding faulty data are insufficient for training purposes. Furthermore, collecting new fault labelled industrial data is expensive and time-consuming. When a given chemical plant operates in a new mode, the model trained using the data collected from the previous mode cannot be effectively used as a fault detection model. This phenomenon occurs due to the differences between the data distributions of the new and previously known modes. Furthermore, due to the limited labelled faulty data in the new mode, it is difficult to redevelop a fault detection model for the new mode. Therefore, the aforementioned techniques, which make the assumption that the training and testing data are drawn from the same input feature space with the same distributions, cannot effectively be applied for this scenario.

Transfer learning is a promising approach to address the challenges associated with the lack of data in some modes, wherein the training data are collected from one mode, and the test data are collected from another with different feature spaces and/or data distributions [12]–[15]. Therefore, transfer learning has been applied in process monitoring applications by transferring the knowledge of the modes with sufficient labelled fault data (i.e., source mode) to the modes with insufficient labelled fault data (i.e., target mode) [16], [17]. Feature-based transfer learning is a commonly used form of transfer learning. The intuitive idea behind feature-based transfer learning is to transfer the source and target features into a common feature space by learning a pair of feature mapping functions. With this approach, the predictions in the target space can be significantly improved as the model is built using features that are common to different modes. Several approaches exist for learning either domain invariant features [18], [19] or universal features, which are common to all modes [20]. Pan introduced transfer component analysis (TCA) wherein some of the features across domains are learnt by minimising maximum mean discrepancy (MMD) in a reproducing kernel Hilbert space [19], [21].

In this paper, a new model based on transfer learning is developed and proposed for process monitoring applications. The proposed model is inspired by [22] and [23] achieving efficient transfer across modes or domains by extracting the features that are common to different domains. We learn the domain-invariant features by: (1) minimizing the reconstruction error of Probabilistic Principal Component Analysis (PPCA) as a feature extractor that is shared across the source and target domains and (2) maximizing the domain classification error of logistic regression as domain classifier. The whole learning process is accomplished under this architecture which we call Domain Adversarial Probabilistic Principal Component Analysis (DAPPCA). The main contributions of this work are: First, a novel model structure, DAPPCA, is proposed for monitoring purposes. The proposed model is composed of probabilistic principal component analysis (PPCA) as a feature extractor and logistic regression as a domain classifier. This structure achieves efficient transfer across domains by extracting the features that cannot discriminate the domain of origin of observation. The considered model would address scenarios like the problem of process monitoring when there is a lack of labelled faulty data in some modes. Second, to learn the domain-invariant features, we jointly optimize two objectives: (1) minimizing the reconstruction error of probabilistic principal component analysis (PPCA) as a feature extractor that is shared across the source and target domains and (2) maximizing the domain classification error of logistic regression as domain classifier. Third, we presented an approach based on the recent advances in variational inference to learn the parameters of the whole architecture. Lastly, we have extensively validated our results with numerical data and real industrial data.

The rest of this paper is organized as follows: Section 2 revisits PPCA as a fault detection tool. Section 3 introduces DAPPCA and an algorithm to learn the parameters of the model using variational inference. This section also presents an algorithm to monitor control charts based on the proposed model. In section 4, a simulation example and an industrial case study are presented to verify the efficiency of the proposed approach. A Summary of our findings and conclusions are provided in section 5.

II. Probabilistic Principal Component Analysis

The PPCA, a probabilistic counterpart of the traditional PCA, tries to project observed data onto a low dimensional feature space using a probabilistic framework. Let us suppose that we observe a d-dimensional data set $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}]^T \in \mathbb{R}^{n\times d}$. Assume that the observed data are generated by an underlying q-dimensional latent variable denoted by $\boldsymbol{\phi}$, where the linear relationship between the observed data and latent variable for each sample i under PPCA framework is defined as,

$$\mathbf{x_i} = \mathbf{v}\boldsymbol{\phi_i} + \epsilon_i \tag{1}$$

where $\mathbf{v} \in \mathbb{R}^{d\times q}$ is the loading matrix variable and the noise variable, ϵ_i, is assumed to follow a Gaussian distribution with zero mean and variance σ^2. In Eq.(1) we assume that the latent variables $\boldsymbol{\phi_i} \in \mathbb{R}^{q\times 1}$ follow a normal distribution denoted by $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The parameters of the PPCA model $\theta_{ppca} = [\mathbf{v}, \sigma^2]$, can be estimated using the Expectation-Maximization (EM) algorithm. The estimated value of θ_{ppca} is obtained by iteratively going through an expectation step (E-step) and a maximization step (M-step). In the E-step, the posterior distribution of $\boldsymbol{\phi_i}$ is derived, and in the M-step, the first and second moments of this posterior are utilized to update the model parameters in θ_{ppca}.

After the PPCA model from the normal operating data has been developed, two corresponding monitoring statistics are constructed that are the same as those of PCA, namely, (1) Hotelling's T^2 [24], and (2) Q [25]. These statistics are monitored to check if a newly collected sample falls in the normal operating range. The T^2 statistic is utilized for monitoring the variability in the latent variables. The squared prediction error (SPE) or Q statistic monitors the space of model residuals. Using the results from the T^2 statistic, we are able to detect the samples where the latent variables drift further away from the origin and from the Q statistic, we are able to detect the samples where the model residuals go out of the desired bounds. The T^2 statistic is the normalized sum of squares of latent variables defined as,

$$T_i^2 = \boldsymbol{\phi_i}^T(\sigma^2\mathbf{I})^{-1}\boldsymbol{\phi_i}. \tag{2}$$

In this work, T^2 statistic is utilized for the purpose of process monitoring.

III. Proposed method

In real industrial settings, often faulty data are limited, and this is particularly true when a process is operating in a new mode. In addition, due to the differences between the distributions of data from new and previously known modes, the model that has been trained using the data collected from the known modes cannot be used to detect faults in the new mode. In this section, we propose a novel DAPPCA algorithm to improve the fault detection model in the new mode by transferring the knowledge from known modes and developing corresponding process monitoring statistics. The proposed transfer learning algorithm involves extracting the features that are shared between the new mode (i.e., source domain) and known modes (i.e., target domain). The terms mode and domain are used interchangeably in the rest of this paper.

A. Domain Adversarial Probabilistic Principal Component Analysis

To implement transfer learning in the context of process monitoring, the architecture illustrated in Fig. (1) is proposed. In this setting, we denote the observed points in the source and target domains by $\mathbf{X_S}$ and $\mathbf{X_T}$ respectively as follows,

$$\begin{aligned}\mathbf{X_S} &= \{\mathbf{x_{S_i}} | i = 1, 2, \ldots, n_S\} \\ \mathbf{X_T} &= \{\mathbf{x_{T_i}} | i = 1, 2, \ldots, n_T\}\end{aligned} \tag{3}$$

where $n = n_S + n_T$ is the total number of source and target sample points.

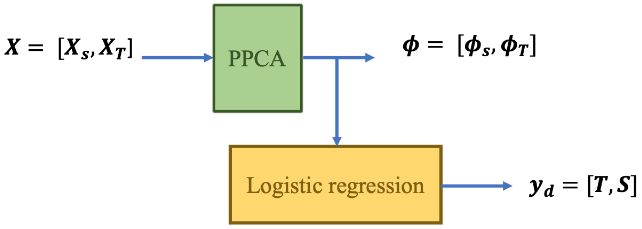

$\boldsymbol{X} = [\boldsymbol{X_s}, \boldsymbol{X_T}]$
PPCA
$\boldsymbol{\phi} = [\boldsymbol{\phi_s}, \boldsymbol{\phi_T}]$
Logistic regression
$\boldsymbol{y_d} = [\boldsymbol{T}, \boldsymbol{S}]$

Fig. 1: A schematic of the proposed method for process monitoring via Transfer learning

The q-dimensional latent variable corresponding to the observed data is denoted by $\boldsymbol{\phi}$ with subscript S for source domain and subscript T for the target domain. Binary outputs, $\mathbf{y_d} = [\mathbf{y_d}_1, \mathbf{y_d}_2, ..., \mathbf{y_d}_n]$ indicate the vector of domain labels corresponding to the observed data, where $\mathbf{y_d}_i \in [T, S]$ is domain label at the i^{th} sample point.

The architecture shown in Fig. (1) consists of two essential parts: (i) a feature extractor, which is shared across both source and target domains and (ii) a domain classifier that discriminates the domain of origin of features. To make feature distribution domain-invariant, the parameters of the feature extractor are optimized in order to maximize the loss of classifying domain labels. By doing so, we obtain domain-invariant features across both source and target domains, namely the features that cannot be distinguished between source and target domain. As a result, the DAPPCA has the model structure given below,

$$\mathbf{x_i} = \mathbf{v}\boldsymbol{\phi_i} + \epsilon_i \tag{4}$$

$$p(\mathbf{y_d}_i|\boldsymbol{\phi_i}) = \Phi(\mathbf{w}^T\boldsymbol{\phi_i}) \tag{5}$$

where $\mathbf{w} \in \mathbb{R}^{q\times 1}$ is the parameter vector of sigmoidal function, Φ. We utilize logistic regression, Φ, to model the probabilities of source and target.

The first goal of this architecture is to learn common feature spaces. Therefore, both source $\mathbf{X_S}$ and target domain data $\mathbf{X_T}$ are projected onto a shared latent feature space $\boldsymbol{\phi}$. To this end, as shown in Eq. (4), a probabilistic approach is taken to define a mapping function from the latent feature space $\boldsymbol{\phi} \in \mathbb{R}^{n\times q}$ to the observation space $\mathbf{X} = [\mathbf{X_S}; \mathbf{X_T}] \in \mathbb{R}^{n\times d}$. This function allows for extracting useful features through dimensionality reduction where $q \ll d$. Furthermore, we train the parameters of the feature extractor such that the features cannot be distinguished between source and target domain by the best classifier. The second goal of this architecture is to train the classifier model in Eq. (5).

Let us concatenate the source and target data for training as $\mathbf{y_d} = [\mathbf{y_d}_\mathbf{S}; \mathbf{y_d}_\mathbf{T}]$, $\mathbf{X} = [\mathbf{x_S}; \mathbf{x_T}]$ and denote the complete set of feature extractor and classifier parameters with $\vartheta = [\underbrace{\mathbf{v}, \sigma_2}_{\theta_{ppca}}, \underbrace{\theta_w, \theta_\phi}_{\theta_{cl}}]$. The overall objective function for the proposed approach can be written as follows:

$$\mathcal{L} = \mathcal{L}_{\mathbf{rec}}(\mathbf{X}, \hat{\mathbf{X}}) - \lambda\mathcal{L}(\mathbf{y_d}, \hat{\mathbf{y}}_\mathbf{d}) \tag{6}$$

where $\mathcal{L}_{\mathbf{rec}}(\mathbf{X}, \hat{\mathbf{X}})$ is the feature extraction loss, $\mathcal{L}(\mathbf{y_d}, \hat{\mathbf{y_d}})$ is the domain classifier loss and λ is a positive scalar trade-off hyper-parameter that appropriately weighs contribution of each loss term. The parameters of feature extractor, PPCA, are optimized in order to minimize feature extraction loss, $\mathcal{L}_{\mathbf{rec}}(\mathbf{X}, \hat{\mathbf{X}})$, and maximize the domain classifier loss, $\mathcal{L}(\mathbf{y_d}, \hat{\mathbf{y_d}})$. By minimizing the objective in Eq. (6) with respect to parameters of feature extractor, the domain-invariant features emerge. This objective can be represented using the corresponding log-likelihood functions as shown below:

$$\mathcal{L} = \Big[\log p(\mathbf{X}|\vartheta) - \lambda \log p(\mathbf{y_d}, \mathbf{X}|\vartheta)\Big] \tag{7}$$

and it can be expanded using marginalization and Chain rule of probability,

$$\begin{aligned}\mathcal{L} = &\Big[\log \int_\phi p(\mathbf{X}, \boldsymbol{\phi}|\vartheta)\ d\phi \qquad (8)\\ &- \lambda \log \int_w \int_\phi p(\mathbf{y_d}, \mathbf{X}, \boldsymbol{\phi}, \mathbf{w}|\vartheta)\ d\boldsymbol{\phi}\ d\mathbf{w}\Big] \\ = &\Big[\log \int_\phi p(\mathbf{X}|\boldsymbol{\phi}, \vartheta)p(\boldsymbol{\phi}|\vartheta)\ d\phi \\ &- \lambda \log \int_w \int_\phi p(\mathbf{y_d}|\boldsymbol{\phi}, \mathbf{w}, \vartheta)p(\mathbf{X}|\boldsymbol{\phi}, \vartheta)p(\boldsymbol{\phi}|\vartheta)p(\mathbf{w}|\vartheta)\ d\phi\ d\mathbf{w}\Big] \qquad (9)\end{aligned}$$

In order to optimize the objective in Eq. (6), we need to compute the log marginal likelihood of data in Eq. (9). However, this integration is intractable. Instead, we propose a variational inference procedure to evaluate the objective function. This approach involves finding a lower bound of the objective function. Given that we want to maximize the first term in Eq. (9), the objective can be altered by replacing it with the evidence lower bound (ELBO) of the log marginal likelihood of data [26]. Similarly, given that we want to minimize the second term in Eq. (9) w.r.t θ_{ppca}, it can be replaced by the evidence upper bound.

The work by [27] provides an upper bound, which is referred to as the χ upper bound (CUBO) that is a special case of Rényi divergence introduced by [28], [29]. In their work, a tractable objective function by χ^2-divergence is derived as in Eq.(10) and this quantity is proved to be a general upper bound to the model evidence.

$$\begin{aligned}&\log p(\mathbf{y_d}, \mathbf{X}) \leq CUBO \\ &where \quad CUBO = \frac{1}{2}\log \mathbb{E}_{q'}[(\frac{p(\mathbf{y_d}, \mathbf{X})}{q'})^2]\end{aligned} \tag{10}$$

q' is a variational distribution family. Using ELBO and CUBO, we can express a total lower bound on Eq. (9) that takes the form as below,

$$\mathcal{F}(q,q',\boldsymbol{\varphi}) = \int q \log \frac{p(\mathbf{X}|\boldsymbol{\phi},\vartheta)p(\boldsymbol{\phi}|\vartheta)}{q}\ d\phi \\ - \frac{\lambda}{2}\log \int q'(\frac{p(\mathbf{y_d}|\boldsymbol{\phi},\mathbf{w},\vartheta)p(\mathbf{X}|\boldsymbol{\phi},\vartheta)p(\boldsymbol{\phi}|\vartheta)p(\mathbf{w}|\vartheta)}{q'})^2\ d\phi\ d\mathbf{w} \tag{11}$$

We now choose the two variational distributions on the latent variables, q and q', to be of the forms below,

$$q(\boldsymbol{\phi}) = p(\boldsymbol{\phi}|\mathbf{X},\vartheta) \tag{12}$$

$$q'(\boldsymbol{\phi},\mathbf{w}) = q'(\mathbf{w};\theta_w)q'(\boldsymbol{\phi};\theta_\phi) \tag{13}$$

The integration in the second term of Eq. (11) is still intractable. Hence, the exponentiated upper bound, $\exp\{2 \times \text{CUBO}\}$, is substituted for the CUBO to produce the same optimum points that are not biased estimates [27].

$$\mathcal{F}(q,q',\boldsymbol{\varphi}) = \mathbb{E}_{p(\boldsymbol{\phi}|\mathbf{X},\vartheta)} \log[p(\mathbf{X}|\boldsymbol{\phi},\vartheta)p(\boldsymbol{\phi}|\vartheta)] + \mathbb{H}(\phi) \\ - \lambda \int q'(\mathbf{w})q'(\boldsymbol{\phi})(\frac{p(\mathbf{y_d}|\boldsymbol{\phi},\mathbf{w},\vartheta)p(\mathbf{X}|\boldsymbol{\phi},\vartheta)p(\boldsymbol{\phi}|\vartheta)p(\mathbf{w}|\vartheta)}{q'(\mathbf{w})q'(\boldsymbol{\phi})})^2 \\ d\phi\ d\mathbf{w} \tag{14}$$

where $\mathbb{H}(\phi) = -q\log q$ is the entropy. Hence, the first term and the third term in the above equation are the only terms to be optimized, and entropy is omitted,

$$\mathcal{F}(q,q',\boldsymbol{\varphi}) = \underbrace{\mathbb{E}_{p(\boldsymbol{\phi}|\mathbf{X},\vartheta)} \log[p(\mathbf{X}|\boldsymbol{\phi},\vartheta)p(\boldsymbol{\phi}|\vartheta)]}_{\mathcal{F}_1} \\ \underbrace{- \lambda\, \mathbb{E}_{q'(\mathbf{w})q'(\boldsymbol{\phi})}(\frac{p(\mathbf{y_d}|\boldsymbol{\phi},\mathbf{w},\vartheta)p(\mathbf{X}|\boldsymbol{\phi},\vartheta)p(\boldsymbol{\phi}|\vartheta)p(\mathbf{w}|\vartheta)}{q'(\mathbf{w})q'(\boldsymbol{\phi})})^2}_{\mathcal{F}_2} \tag{15}$$

First, the lower bound $\mathcal{F}_1$ is derived. In PPCA, the posterior distribution of the latent variable, $p(\boldsymbol{\phi}_i|\mathbf{x}_i;\mathbf{v},\sigma^2)$, is Gaussian and can be calculated based on Bayes' rule as,

$$p(\boldsymbol{\phi}_i|\mathbf{x}_i;\mathbf{v},\sigma^2) \\ = \mathcal{N}(\boldsymbol{\phi}_i|(\mathbf{v}^T\mathbf{v}+\sigma^2 I)^{-1}\mathbf{v}^T\mathbf{x}_i, \sigma^2(\mathbf{v}^T\mathbf{v}+\sigma^2\mathbf{I})^{-1}) \tag{16}$$

Having obtained the posterior distribution of $p(\boldsymbol{\phi}_i|\mathbf{x}_i;\mathbf{v},\sigma^2)$, the final form of lower bound $\mathcal{F}_1$ after taking expectation is given as below,

$$\mathcal{F}_1 = \mathbb{E}_{p(\boldsymbol{\phi}_i|\mathbf{x}_i)} \log \prod_i^N [p(\mathbf{x}_i|\boldsymbol{\phi}_i,\vartheta)p(\boldsymbol{\phi}_i|\vartheta)] \\ = \mathbb{E}_{p(\boldsymbol{\phi}_i|\mathbf{x}_i)} \sum_i^N \log[\mathcal{N}(\mathbf{x}_i|\mathbf{v}\boldsymbol{\phi}_i,\sigma^2\mathbf{I})\mathcal{N}(\boldsymbol{\phi}_i|\mathbf{0},\mathbf{I})] \\ = \sum_i^N \{ -\frac{D}{2}\log 2\pi\sigma^2 - \frac{\mathbf{x}_i^T\mathbf{x}_i}{2\sigma^2} + \frac{\mathbb{E}_{p(\boldsymbol{\phi}_i|\mathbf{x}_i)}[\boldsymbol{\phi}_i]^T\mathbf{v}^T\mathbf{x}_i}{\sigma^2} \\ - \frac{Tr(\mathbb{E}_{p(\boldsymbol{\phi}_i|\mathbf{x}_i)}[\boldsymbol{\phi}_i\boldsymbol{\phi}_i^T]\mathbf{v}^T\mathbf{v})}{2\sigma^2} - \frac{D}{2}\log 2\pi \\ - \frac{Tr(\mathbb{E}_{p(\boldsymbol{\phi}_i|\mathbf{x}_i)}[\boldsymbol{\phi}_i\boldsymbol{\phi}_i^T])}{2}\} \tag{17}$$

Next, the upper bound $\mathcal{F}_2$ is minimized using CHIVI algorithm [27], [30]. We restrict two variational distributions $q'(\boldsymbol{\phi};\theta_\phi)$ and $q'(\boldsymbol{w};\theta_w)$ to be Gaussians. Given variational distributions $q'(\boldsymbol{\phi};\theta_\phi)$ and $q'(\boldsymbol{w};\theta_w)$, the gradients of CUBO can be derived. First, we integrate out θ_w and derive the gradient of $\theta_\phi = [\mu_\phi, \sigma_\phi^2]$ as follows,

$$\frac{\partial\mathcal{F}_2}{\partial\theta_\phi} = -\mathbb{E}_{q'(\boldsymbol{\phi};\theta_\phi)}\Big[\mathbb{E}_{q'(\boldsymbol{w};\theta_w)}[\\ (\frac{p(\mathbf{y_d}|\boldsymbol{\phi},\mathbf{w})p(\mathbf{X}|\boldsymbol{\phi})p(\boldsymbol{\phi})p(\mathbf{w})}{q'(\mathbf{w})q'(\boldsymbol{\phi})})^2]\ \nabla_{\theta_\phi}\log q'(\boldsymbol{\phi};\theta_\phi)\Big] \tag{18}$$

Similarly, we derive the gradient of $\theta_w = [\mu_w, \sigma_w^2]$ and integrate out θ_ϕ,

$$\frac{\partial\mathcal{F}_2}{\partial\theta_w} = -\mathbb{E}_{q'(\boldsymbol{w};\theta_w)}\Big[\mathbb{E}_{q'(\boldsymbol{\phi};\theta_\phi)}[\\ (\frac{p(\mathbf{y_d}|\boldsymbol{\phi},\mathbf{w})p(\mathbf{X}|\boldsymbol{\phi})p(\boldsymbol{\phi})p(\mathbf{w})}{q'(\mathbf{w})q'(\boldsymbol{\phi})})^2]\ \nabla_{\theta_w}\log q'(\boldsymbol{w};\theta_w)\Big] \tag{19}$$

The gradient of the lower bound $\mathcal{F}_2$ with respect to feature extraction parameters, $\theta_{ppca} = [\mathbf{v},\sigma^2]$, can be derived as below,

$$\frac{\partial\mathcal{F}_2}{\partial\theta_{ppca}} \\ = \nabla_{\theta_{ppca}} \int q'(\mathbf{w})q'(\boldsymbol{\phi})(\frac{p(\mathbf{y_d}|\boldsymbol{\phi},\mathbf{w})p(\mathbf{X}|\boldsymbol{\phi};\theta_{ppca})p(\boldsymbol{\phi})p(\mathbf{w})}{q'(\mathbf{w})q'(\boldsymbol{\phi})})^2\ d\phi\ d\mathbf{w} \\ = \int q'(\mathbf{w})q'(\boldsymbol{\phi})(\frac{p(\mathbf{y_d}|\boldsymbol{\phi},\mathbf{w})p(\boldsymbol{\phi})p(\mathbf{w})}{q'(\mathbf{w})q'(\boldsymbol{\phi})})^2\nabla_{\theta_{ppca}}p(\mathbf{X}|\boldsymbol{\phi};\theta_{ppca})^2\ d\phi\ d\mathbf{w} \\ = \int q'(\mathbf{w})q'(\boldsymbol{\phi})(\frac{p(\mathbf{y_d}|\boldsymbol{\phi},\mathbf{w})p(\boldsymbol{\phi})p(\mathbf{w})}{q'(\mathbf{w})q'(\boldsymbol{\phi})})^2 \\ 2\ p(\mathbf{X}|\boldsymbol{\phi};\theta_{ppca})\nabla_{\theta_{ppca}}p(\mathbf{X}|\boldsymbol{\phi};\theta_{ppca})\ d\phi\ d\mathbf{w} \\ = 2\int q'(\mathbf{w})q'(\boldsymbol{\phi})(\frac{p(\mathbf{y_d}|\boldsymbol{\phi},\mathbf{w})p(\boldsymbol{\phi})p(\mathbf{w})}{q'(\mathbf{w})q'(\boldsymbol{\phi})})^2 \\ p(\mathbf{X}|\boldsymbol{\phi};\theta_{ppca})^2\nabla_{\theta_{ppca}}\log p(\mathbf{X}|\boldsymbol{\phi};\theta_{ppca})\ d\phi\ d\mathbf{w} \\ = 2\ \mathbb{E}_{q'(\boldsymbol{w};\theta_w)}\Big[\mathbb{E}_{q'(\boldsymbol{\phi};\theta_\phi)}[\\ (\frac{p(\mathbf{y_d}|\boldsymbol{\phi},\mathbf{w})p(\mathbf{X}|\boldsymbol{\phi})p(\boldsymbol{\phi})p(\mathbf{w})}{q'(\mathbf{w})q'(\boldsymbol{\phi})})^2\ \nabla_{\theta_{ppca}}\log p(\mathbf{X}|\boldsymbol{\phi},\theta_{ppca})]\Big] \tag{20}$$

The expectation terms in gradient equations (18), (19), and (20) are estimated by Monte Carlo estimation. Ultimately, the update equations for parameters of features extractor and domain classifier using the above gradient equations can be obtained as in Eqs. (21) to (26).

Note that there is a subtraction between the gradients of the lower bound and the upper bound for updating the parameters of the feature extractor in Eqs. (21) and (23), owing to the fact that we wish to maximize the domain classification loss to extract the domain-invariant features from source and target domains.

$$\mathbf{v}_{new} = \mathbf{v} - \eta[\frac{\partial\mathcal{F}_1}{\partial\mathbf{v}} - \lambda\frac{\partial\mathcal{F}_2}{\partial\mathbf{v}}] \tag{21}$$

$$\boldsymbol{\sigma^2}_{new} = \boldsymbol{\sigma^2} - \eta[\frac{\partial\mathcal{F}_1}{\partial\sigma^2} - \lambda\frac{\partial\mathcal{F}_2}{\partial\sigma^2}] \tag{22}$$

where η denotes the learning rate schedule.

Besides, on the grounds that we wish to find the optimal classifier, we minimize the domain classification loss w.r.t parameters of the classifier in Eqs. (23) to (26). We first update the parameters of the feature extractor part. Next, the parameters of the domain classifier are updated. Repeat these two steps to train the feature extractor and domain classifier.

$$\mu_{\phi new} = \mu_\phi - \eta[\lambda \frac{\partial \mathcal{F}_2}{\partial \mu_\phi}] \tag{23}$$

$$\sigma^2_{\phi new} = \sigma^2_\phi - \eta[\lambda \frac{\partial \mathcal{F}_2}{\partial \sigma^2_\phi}] \tag{24}$$

$$\mu_{w new} = \mu_w - \eta[\lambda \frac{\partial \mathcal{F}_2}{\partial \mu_w}] \tag{25}$$

$$\sigma^2_{w new} = \sigma^2_w - \eta[\lambda \frac{\partial \mathcal{F}_2}{\partial \sigma^2_w}] \tag{26}$$

B. Monitoring strategy based on DAPPCA

For fault detection analysis based on the DAPPCA model developed from normal operating data, we derive the commonly used statistic T^2. Given the posterior distribution of $p(\phi_i|\mathbf{x}_i, \mathbf{y_d}_i) \sim \mathcal{N}(\phi_i|\mu_\phi, \sigma^2_\phi)$, T^2 statistic is similarly derived as in section (II) for sample i,

$$\mathbf{T_i^2} = \boldsymbol{\phi_i}^T.(\sigma^2_\phi)^{-1}.\boldsymbol{\phi_i} \tag{27}$$

The flowchart of the DAPPCA monitoring algorithm is shown in Fig. (8)

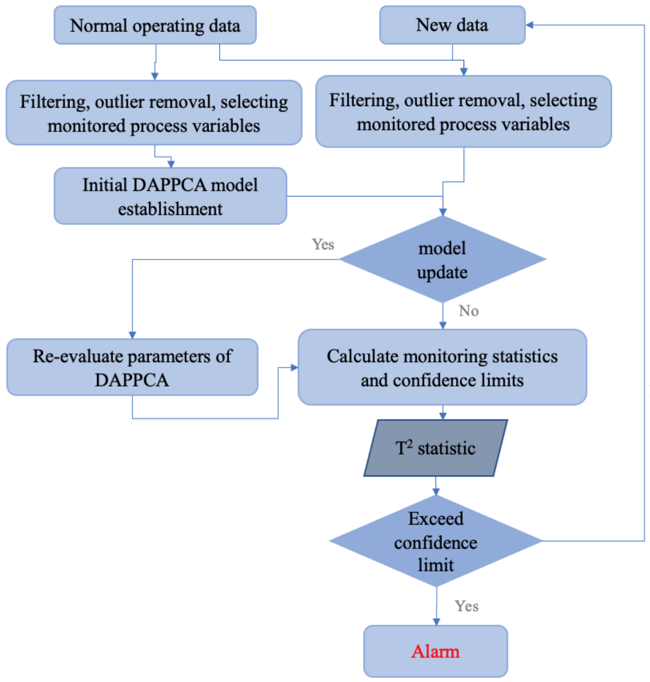

Normal operating data
New data
Filtering, outlier removal, selecting monitored process variables
Filtering, outlier removal, selecting monitored process variables
Initial DAPPCA model establishment
Yes
model update
No
Re-evaluate parameters of DAPPCA
Calculate monitoring statistics and confidence limits
No
T² statistic
Exceed confidence limit
Yes
Alarm

Fig. 2: The DAPPCA process monitoring strategy flow chart

IV. Case study

A. Simulated example

To evaluate the effectiveness of our proposed DAPPCA method, a synthetic dataset is first generated. More specifically, the two-dimensional latent features $\boldsymbol{\phi}$ are generated from a Gaussian distribution with zero mean and unit variance. In this synthetic dataset, 150 samples are used as source samples, and their corresponding domain labels, $\mathbf{y_d}$, are labelled as "0", and 240 samples are utilized as the target samples (40 as training and 200 for testing)and their corresponding domain labels, $\mathbf{y_d}$, are labelled as "1". Then, two different random 6×2 matrices $\mathbf{v_s}$ and $\mathbf{v_T}$ are constructed. Using the generated $\mathbf{v}$ and $\boldsymbol{\phi}$ for both source and target domains, the source and target data $\mathbf{X_s}$ and $\mathbf{X_T}$ are drawn from a Gaussian distribution with mean $\mathbf{v}\boldsymbol{\phi}^T$ and the variance 0.01. We then investigated the transferring behavior of DAPPCA in extracting the common features across domains by comparing it with regular PPCA. To implement DAPPCA, we used the AdaGrad algorithm and set the learning rate to 1. For gradient estimations of the CUBO part, ten samples are utilized at each iteration.

The estimation results of latent features for 40 samples of target data are presented in Fig.3. As it is evident from the results that the proposed method outperforms the PPCA and mixture probabilistic principal component (MPPCA) method in terms of estimation performance. The main reason why PPCA is not capturing the feature representation well is due to limited available target data.

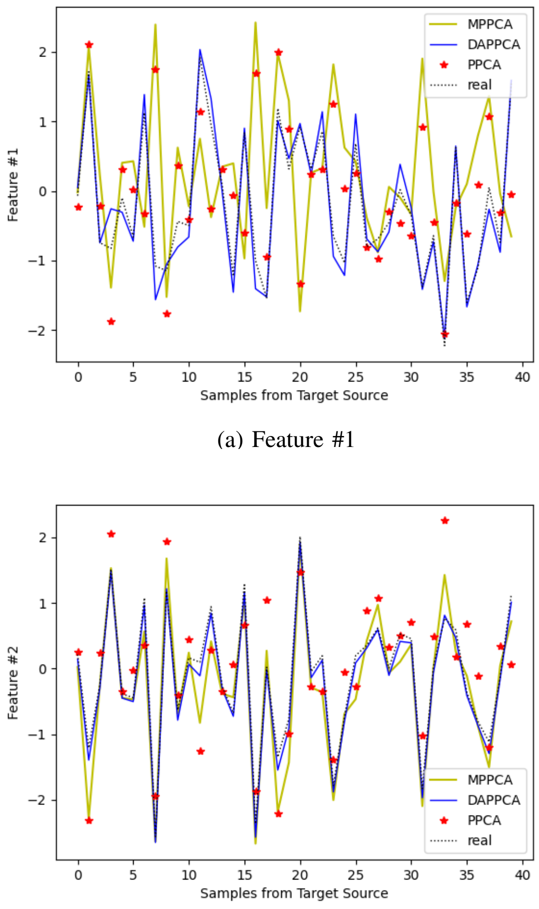
Furthermore, Table I shows the magnitudes of concordance correlation coefficients and mean absolute errors of features using DAPPCA, MPPCA, and PPCA methods. The concordance correlation coefficient between real features and predictions by the proposed method is higher than the existing methods, i.e., it shows the actual value of features and the predicted ones are concordant. Hence, the magnitudes of concordance correlation coefficients and mean absolute error clearly indicate the superiority of the proposed approach in the estimation of the domain-invariant features.

	DAPPCA		PPCA		MPPCA	
	ρ_S	MAE	ρ_S	MAE	ρ_S	MAE
Feature #1	0.9757	0.1767	0.3521	0.8686	0.3419	0.8287
Feature #2	0.9558	0.0944	0.8038	0.5553	0.8856	0.3025

TABLE I: Comparison of concordance correlation coefficients and mean absolute error of the latent features

Fig. 4 represents the change in variational means of $q(\phi)$ at every iteration. From the figure, it can be observed that the change in variational mean drops during the first iterations. Afterwards, it remains almost constant with an increase in the number of iterations.

To test the algorithm performance for monitoring purpose, some faulty data are introduced with a step change with amplitude 10 from the 140th to 240th points in the first dimension of the target data. The monitoring results using the DAPPCA, PPCA, mixture PPCA (MPPCA), recursive PPCA (RPPCA), and probabilistic slow feature analysis (PSFA) methods are provided for comparison. The monitoring results of these methods for fault in target data are illustrated in Fig. 5(a)-(e). Fig. 5 compares the T^2 statistic obtained from these methods. The control limit of T^2 is determined by the χ^2 distribution



(a) Feature #1

(b) Feature #2

Fig. 3: Estimation of latent variables with the proposed method. Real represents the original features, DAPPCA, MPPCA, and PPCA represent the estimation of the features with the proposed method, MPPCA, and PPCA, respectively.

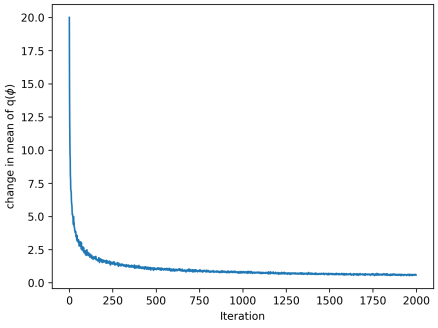


Fig. 4: Change in mean of $q(\phi)$

with a confidence level of 95 % .

To statistically characterize the performance of the presented process monitoring method, we employ two metrics, namely, fault detection rate (FDR) and false alarm rate (FAR), defined as follows:

$$FDR = \frac{Detection}{Fault} \qquad FAR = \frac{False\ alarms}{No\ fault} \tag{28}$$

As it is evident from Fig. 5 and Table II, the proposed method provides superior monitoring performance compared to other methods, with better detection and false alarm rates. The MPPCA monitoring results do not indicate the fault but PPCA, RPPCA and PSFA detect the fault with a worse detection rate, false alarm rate, and detection delay than that of DAPPCA.

	DAPPCA		PPCA		MPPCA		RPPCA		PSFA	
	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR	FDR	FAR
Target	100%	1.43%	35%	2.14%	0%	1.43%	32%	6.43%	100%	12.86%

TABLE II: Comparison of different methods in terms of the fault detection rate and false alarm rate

In the following section, we demonstrate the performance of the proposed algorithm on an industrial data set.

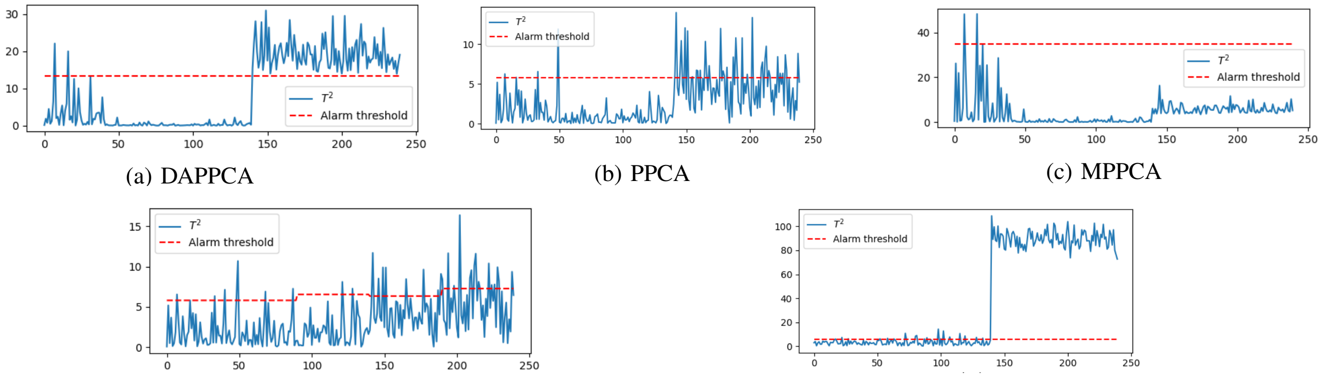
B. Electrical submersible pump (ESP)

The effectiveness of our proposed DAPPCA method is further investigated through an industrial monitoring problem.

Electrical submersible pumps (ESPs) are the predominant artificial lifting systems that are widely utilized in upstream oil production. Therefore, monitoring and detection of ESP failures is vital to avoid ESP failures and the resulting lifting costs and loss of production from downtime. In this section, the proposed monitoring algorithm is deployed to monitor the long-term reliability of the ESP operation. To this end, the most informative process variables that are to be used for monitoring algorithm design are selected. The process schematic of the steam-assisted gravity drainage (SAGD) process with the critical variables is illustrated in Fig.6. We consider measurements of the six most informative process variables, including pump frequency, motor current, bottom hole temperatures, and production tubing temperature, for monitoring ESP performance.

Data pre-processing as one of the important stages of process monitoring is applied to the raw data to remove noise and other disturbances. The data pre-processing involves filtering the data to remove outliers and unwanted peaks in process variables during process shutdown. The filtering process is performed by substituting each sample by the median of samples from a fixed size window. Next, outlier removal is performed by removing the data out of the expected operating range. The development of a soft sensor for motor current prediction is crucial for successful online monitoring of the ESP performance. Therefore a soft sensor is developed to predict motor current using a linear regression model, which relates the square of the pump frequency to the motor current. The soft sensor prediction error is treated as one of the monitored variables. It is calculated using the predicted motor current obtained by the soft sensor and the actual motor current measurements. In addition, the temperature differences and temperature gradients are also treated as monitored variables. A list of all the monitored variables for ESPs is shown in Table III.

After the pre-processing stage, the initial statistical feature extraction model is trained on the monitored variables to be



(a) DAPPCA

(b) PPCA

(c) MPPCA

(d) RPPCA

(e) PSFA

Fig. 5: Monitoring results of (a) DAPPCA, (b) PPCA, (c) MPPCA, (d) RPPCA, and (e) PSFA for fault in simulated data

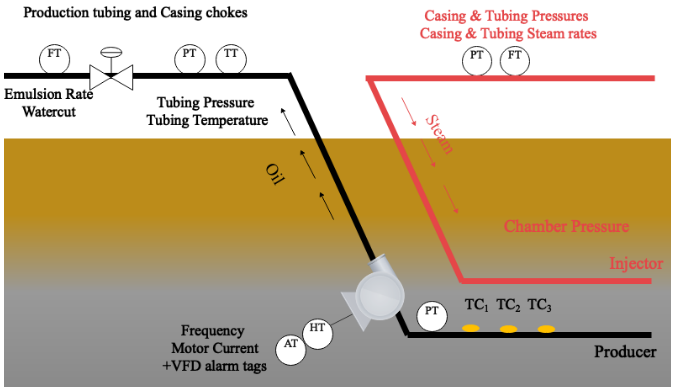


Fig. 6: A Schematic diagram of SAGD process variables of interest

Collected process variables	Monitored process variables
Bottom hole temperatures: $T_i, i \in [1,2,3]$ Production tubing temperature: T	Temperature gradients: $\frac{T_i - T}{T_i}, i \in [1,2,3]$
Bottom hole temperatures: $T_i, i \in [1,2,3]$	Temperature differences: $T_i - T_{i+1}, i \in [1,2]$
Motor current: I Pump Frequency: F	Predicted $I \propto F^2$ $(I - PredictedI)/PredictedI$

TABLE III: Monitored variables used in ESP Monitoring

used for the subsequent samples to perform online monitoring of the ESP performance. Finally, T^2 monitoring statistic is generated to detect the failures in advance.

The proposed transfer learning-based method is used for the detection of a failure in ESP. To demonstrate the performance of this method, the historical data from one failed ESP is considered as source data. The data collected from another ESP is considered as target data. All the failure portions are removed from the data so that any abrupt change in T^2 at the end of the data can be considered as early detection of ESP failure. While building an initial model using the proposed method, input variables to the algorithm are constructed by 6000 samples before the end of the source data set along with the 2640 samples from the beginning of target data. Afterwards, the motor current soft sensor model, DAPPCA model, are learned by these samples. The parameters of the ESP soft sensor and DAPPCA are used to perform monitoring on subsequent data from the target ESP to predict its failure in advance. By using the proposed algorithm, for the target ESP case, the monitoring result is depicted in Fig. 7 as follows,

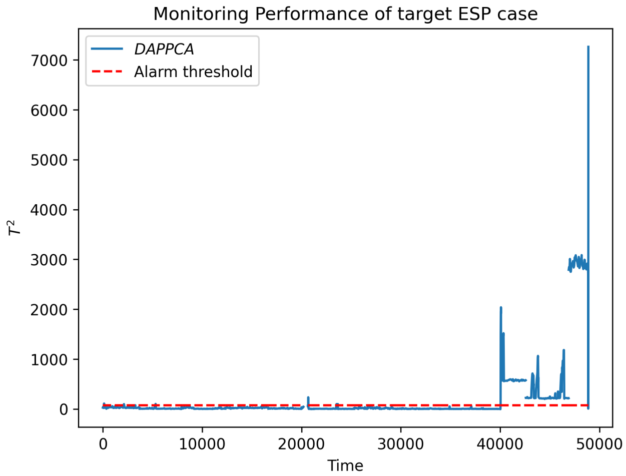


Fig. 7: Monitoring performance of Target ESP case using DAPPCA

In Fig. 7, the blue line represents T^2 monitoring statistic, and the red dotted line is the control limit. From the monitoring result, it can be observed that there is faulty behavior at the end portion of data which affirms the proposed transfer learning-based method can effectively predict the ESP failure before happening.

Fig. 8 shows the monitoring result on the target data using traditional PPCA (without transferring knowledge from another ESP case).

The early detection and FAR of the proposed method and four existing methods, including PPCA, mixture probabilistic principal component analysis (MPPCA), recursive probabilistic principal component analysis (RPPCA), and probabilistic slow feature analysis (PSFA) are presented in Table (II).

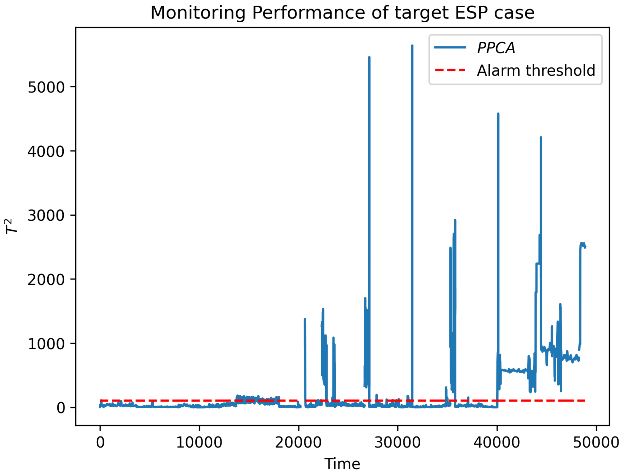


Fig. 8: Monitoring performance of Target ESP case using PPCA

	DAPPCA		PPCA		MPPCA		RPPCA		PSFA	
	Detection	FAR	Detection	FAR	Detection	FAR	Detection	FAR	Detection	FAR
Target ESP	Yes	0.58%	Yes	25.74%	No	4.38%	Yes	11.45%	Yes	32.93%

TABLE IV: Comparison of the proposed method and traditional PPCA in terms of the fault detection and false alarm rate

From the above result, it can be concluded that the proposed DAPPCA method achieves a better detection result compared to other methods in terms of false alarms. The traditional PPCA and other aforementioned methods do not transfer knowledge from the source ESP data. Thus, there are considerable false alarms as shown in Table IV. From Fig. 8 (note the actual failure occurred at the end), even though the traditional PPCA also shows alarms before ESP failure. The proposed transfer learning-based method shows a larger significant value of T^2 monitoring statistic at the end, which demonstrates the early warning of ESP failure. The better performance of the proposed method is attributed to its capability of accounting for the latent features leading to failure of source ESP (across domain). In this way, more information about the extracted features can be included in the proposed method, which can better model the key characteristics of the industrial data and avoid being misled by false alarms caused by other process data variations.

V. Conclusions

In this paper, a domain adversarial probabilistic principal component analysis (DAPPCA) was developed for transferring the knowledge of the modes with sufficient fault labelled data (i.e., source mode) to the modes with limited faulty data (i.e., target mode) for process monitoring purpose. More precisely, the DAPPCA structure was adopted to learn the common feature space to which both source and target data are projected. Hence, the shared fault features improve the accuracy of the fault detection model in the target mode based on the knowledge transferred from the source mode. Further, we presented a variational inference approach to obtain the proposed model parameters. A simulated example was utilized to demonstrate the advantage of the proposed method. In addition, the method was applied to monitor industrial electrical submersible pumps (ESP), which further verified the effectiveness of the proposed approach.

VI. Acknowledgment

This work is supported in part by Natural Science Engineering Research Council of Canada Industrial Research Chair program in Control of Oil Sands Processes and Alberta Innovates. The authors also would like to acknowledge Dr. Mengqi Fang for her helpful advice and insights.

References

[1] J. F. MacGregor and T. Kourti, “Statistical process control of multivariate processes,” *Control Engineering Practice*, vol. 3, no. 3, pp. 403–414, 1995.

[2] S. Joe Qin, “Statistical process monitoring: basics and beyond,” *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 17, no. 8-9, pp. 480–502, 2003.

[3] C. Xia, J. Howell, and N. F. Thornhill, “Detecting and isolating multiple plant-wide oscillations via spectral independent component analysis,” *Automatica*, vol. 41, no. 12, pp. 2067–2075, 2005.

[4] Q. Jiang, X. Yan, and B. Huang, “Performance-driven distributed pca process monitoring based on fault-relevant variable selection and bayesian inference,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 377–386, 2015.

[5] D. Kim and I.-B. Lee, “Process monitoring based on probabilistic pca,” *Chemometrics and intelligent laboratory systems*, vol. 67, no. 2, pp. 109–123, 2003.

[6] Z. Ge and Z. Song, “Mixture bayesian regularization method of ppca for multimode process monitoring,” *AIChE journal*, vol. 56, no. 11, pp. 2838–2849, 2010.

[7] L. Zhou, J. Chen, Z. Song, Z. Ge, and A. Miao, “Probabilistic latent variable regression model for process-quality monitoring,” *Chemical Engineering Science*, vol. 116, pp. 296–305, 2014.

[8] R. Raveendran, H. Kodamana, and B. Huang, “Process monitoring using a generalized probabilistic linear latent variable model,” *Automatica*, vol. 96, pp. 73–83, 2018.

[9] M. Fang, F. Ibrahim, H. Kodamana, B. Huang, N. Bell, and M. Nixon, “Hierarchically distributed monitoring for the early prediction of gas flare events,” *Industrial & Engineering Chemistry Research*, vol. 58, no. 26, pp. 11 352–11 363, 2019.

[10] R. Raveendran and B. Huang, “Conjugate exponential family graphical models in process monitoring: A tutorial review,” *Chemometrics and Intelligent Laboratory Systems*, p. 104095, 2020.

[11] J. Jiang and Q. Jiang, “Variational bayesian probabilistic modeling framework for data-driven distributed process monitoring,” *Control Engineering Practice*, vol. 110, p. 104778, 2021.

[12] S. Shao, S. McAleer, R. Yan, and P. Baldi, “Highly accurate machine fault diagnosis using deep transfer learning,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2446–2455, 2018.

[13] C. Sun, M. Ma, Z. Zhao, S. Tian, R. Yan, and X. Chen, “Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing,” *IEEE transactions on industrial informatics*, vol. 15, no. 4, pp. 2416–2425, 2018.

[14] Z. Chen, K. Gryllias, and W. Li, “Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 339–349, 2019.

[15] M. Alencastre-Miranda, R. M. Johnson, and H. I. Krebs, “Convolutional neural networks and transfer learning for quality inspection of different sugarcane varieties,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 787–794, 2020.

[16] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[17] J. Xie, B. Huang, and S. Dubljevic, “Transfer learning for dynamic feature extraction using variational bayesian inference,” *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[18] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf, "Kernel methods for measuring independence," *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 2075–2129, 2005.

[19] S. J. Pan, J. T. Kwok, Q. Yang *et al.*, "Transfer learning via dimensionality reduction." in *AAAI*, vol. 8, pp. 677–682, 2008.

[20] M. Chen, Z. Xu, K. Q. Weinberger, and F. Sha, "Marginalized stacked denoising autoencoders," in *Proceedings of the Learning Workshop, Utah, UT, USA*, vol. 36, 2012.

[21] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.

[22] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Advances in neural information processing systems*, pp. 137–144, 2007.

[23] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[24] H. Hotelling, "Multivariate quality control," *Techniques of statistical analysis*, 1947.

[25] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associated with principal component analysis," *Technometrics*, vol. 21, no. 3, pp. 341–349, 1979.

[26] D. Lee, J. Su, and S. J. Qiu, "Variational inference," 2020.

[27] A. B. Dieng, D. Tran, R. Ranganath, J. Paisley, and D. Blei, "Variational inference via χ upper bound minimization," in *Advances in Neural Information Processing Systems*, pp. 2732–2741, 2017.

[28] Y. Li and R. E. Turner, "Variational inference with rényi divergence," *stat*, vol. 1050, p. 6, 2016.

[29] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt, "Advances in variational inference," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 2008–2026, 2018.

[30] R. Ranganath, D. Tran, and D. Blei, "Hierarchical variational models," in *International Conference on Machine Learning*, pp. 324–333. PMLR, 2016.

Atefeh Daemi holds two B.Sc. degrees in petroleum and chemical engineering, both awarded from Sharif University of Technology, Iran; and received her M.Sc. degree in process control from the University of Alberta, Canada. She is currently a Ph.D. student and her research works have mainly focused on Bayesian inference, system identification, data analytics, and machine learning.

Bhushan Gopaluni is a professor in the department of chemical and biological engineering and an Associate Dean for Education and Professional Development in the faculty of Applied Science at the University of British Columbia. He is also an associate faculty in the Institute of Applied Mathematics, the Institute for Computing, Information and Cognitive Systems, Pulp and Paper Center and the Clean Energy Research Center. He was the Elizabeth and Leslie Gould Teaching Professor from 2014 to 2017. He is currently an associate editor for Journal of Process Control, The Journal of Franklin Institute and Results in Control and Optimization. Bhushan received a Ph.D. from the University of Alberta in 2003 and a Bachelor of Technology from the Indian Institute of Technology, Madras in 1997 both in the filed of chemical engineering. From 2003 to 2005 he worked as an engineering consultant at Matrikon Inc. (now Honeywell Process Solutions) during which he had designed and commissioned multivariable controllers in British Columbia's pulp and paper industry, and had implemented numerous controller performance monitoring projects in the Oil Gas and other chemical industries. He is the recipient of Killam Teaching Prize and the Dean's service medal from the University of British Columbia and D.G. Fisher Award in Process Control from Canadian Society for Chemical Engineers.

Biao Huang (M'97-SM'11-F'18) obtained his PhD degree in Process Control from the University of Alberta, Canada, in 1997. He also had MSc degree (1986) and BSc degree (1983) in Automatic Control from the Beijing University of Aeronautics and Astronautics. Biao Huang joined the University of Alberta in 1997 as an Assistant Professor in the Department of Chemical and Materials Engineering, and is currently a full Professor, NSERC Industrial Research Chair in Control of Oil Sands Processes, and AITF Industry Chair in Process Control (2013-2018). He is a Fellow of the Canadian Academy of Engineering and Fellow of Chemical Institute of Canada. He is recipient of Germany's Alexander von Humboldt Research Fellowship, Canadian Chemical Engineer Society's Syncrude Canada Innovation and D.G. Fisher awards, APEGAs Summit Research Excellence award, University of Alberta's McCalla and Killam Professorship awards, Petro-Canada Young Innovator Award, AsTech Outstanding Achievement in Science & Engineering Award and a Best Paper Award from Journal of Process Control. Biao Huang's research interests include: data analytics, process control, system identification, control performance assessment, Bayesian methods and state estimation. Biao Huang has applied his expertise extensively in industrial practice.