Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Improved Process Diagnosis Using Fault Contribution Plots from Sparse Autoencoders

Ásgeir Daniel Hallgrímsson * **Hans Henrik Niemann** *
Morten Lind *

* *Automation and Control Group, Dept. of Electrical Engineering,*
Technical University of Denmark, Kgs. Lyngby, 2800, Denmark,
(e-mail: {asdah, hhn, mli}@ elektro.dtu.dk).

Abstract: Development of model-based fault diagnosis methods is a challenge when industrial systems are large and exhibit complex process behavior. Latent projection (LP), a statistical method that extract features of data via dimensionality reduction, is an alternative approach to diagnosis as it can be formulated to not rely on process knowledge. However, LP methods may perform poorly at identifying abnormal process variables due a "fault smearing" effect - variables unaffected by a fault are unintentionally characterized as being abnormal. The effect occurs because data compression permits faulty and non-faulty variables to interact. This paper presents an autoencoder (AE), a nonlinear LP method based on neural networks, as a monitoring method of a simulated nonlinear triple tank process (TTP). Simulated process data was used to train the AE to generate a monitoring statistic representing the condition of the TTP. Sparsity was introduced in the AE to reduce variable interactivity. The AE's ability to detect a fault was demonstrated. The individual contributions of process variables to the AE's monitoring statistic were analyzed to reveal the process variables that were no longer consistent with normal operating conditions. The key result in this study was that sparsity reduced fault smearing onto unaffected variables and increased the contributions of actual faulty variables.

Keywords: Fault detection and isolation, machine learning, grey box modelling, learning for control, subspace methods.

1. INTRODUCTION

Effective online monitoring of process performance is integral for maintaining stable plant operation, maximizing production, and ensuring the survivability of industrial systems. In fact, abnormal events that disrupt plant performance can cause up to 8% annual loss in production profit (Bullemer et al. [2008]). Due to the increasing complexity of large-scale industrial processes, statistical methods - which can be formulated to not rely on process knowledge - are a practical alternative to more traditional and rigorous model-based fault detection methods. The relevance of statistical monitoring schemes is further supported by the current trend of industries to generate industrial big data thanks to the integration of additional sensors, computers, and other technological artifacts connected to every industrial process (Yan et al. [2017]).

This approach to quality control is known as statistical process control (SPC) (MacGregor et al. [1995]). An important component of SPC is diagnosis of a detected abnormal event and determining its cause. Once an unintended plant upset is identified, it is typically up to the operators to decipher which statistical quality variables contain signal characteristics that help diagnose the problem. Unfortunately, industrial application of SPC-based event diagnosis is ineffective since the most common practice for monitoring the quality of a process is to observe traditional univariate control charts such as Schewart, CUSUM, and EWMA (Brooks et al. [2014]). Their application inherently assumes that process variables are independent of one another, potentially making their use ineffective at diagnosing events that affect multiple process variables.

Multivariate quality control (MQC) methods - which produce quality variables that summarize the condition of several process variables - are a better alternative to univariate approaches for monitoring of multivariable processes (Peres et al. [2018]). Essentially, the Hotelling's T^2 and Q statistics are paired with latent projection (LP) - dimensionality reduction methods such as principal component analysis (PCA) that uncover the correlation structure of data - to detect out-of-control situations. A process is monitored by comparing current plant behavior with an LP model representing its "in-control" behavior. An abnormal event that changes the correlation between process variables is detected when the monitored deviation between the current process state and that predicted by the model exceeds a threshold.

Industrial applications of LP-based process monitoring tend to use linear methods, such as PCA, due to their ease of implementation. Unsurprisingly, linear methods result in high Type I and Type II error rates if the process is nonlinear (Hallgrímsson et al. [2019]; Yan et al. [2016]). Nonlinear extensions of PCA have emerged to uncover both linear and nonlinear correlations between variables. The focus of this paper is on autoencoders (AEs); a type of artificial neural network that learns salient, encoded representations via nonlinear transformations of an original data set. Dong and McAvoy [1996] show that AEs can discover principal curves, i.e., a one-dimensional curve whose shape provides a nonlinear summary of the nonlinear structure of the complex data set it passes through. Kramer [1991] demonstrates significant improvement in nonlinear feature extraction by using a multi-layered AE as opposed to a single-layered AE, assuming that the dimension of latent layers were consistent.

Recent advances in AE-based process monitoring have been made by including developments from other applications of neural networks. Yan et al. [2016] observed improved fault detectability of the Tennessee Eastman process over PCA-based process monitoring by using novel variants of AEs; denoising AEs, which reconstruct the uncorrupted version of corrupted input data, and contractive AEs, which penalize the sensitivity of hidden representations to small (noisy) perturbations around the input. Lee et al. [2019] used a variational autoencoder (VAE) to enforce the monitored data to follow a multivariate normal distribution in the latent space to facilitate appropriate use of Hotelling's T^2 monitoring charts for nonlinear and nonnormal processes, resulting in a reduction of Type I and Type II error rates. Osmani et al. [2019] monitored the condition of a turbo-compressor using a recurrent neural network (RNN) that captured temporal dependency of process variables with the additional regularization constraint that activations in the reduced space followed a Bernoulli distribution. Cheng et al. [2019] combined VAEs and RNNs to produce a variational recurrent neural network for fault detection of the Tennessee Eastman process.

Contributions in the AE-based SPC literature tend to prioritize fault detection over fault isolation. Much of the subject matter focuses on reducing Type I and Type II error detection rates by: (a) increasing model sensitivity to faults; (b) obtaining more robust and complex monitoring statistics; and (c) reducing hampering effects from nominal process changes. Though AEs have been used as a pretraining step for fault-classification networks when labeled fault data is scarce (Sun et al. [2016]), few methods exist where fault isolation is performed exclusively with an AE. However, rudimentary diagnosis with PCA models can be carried out with the analysis of fault contribution plots (Joe Qin [2003]; Miller et al. [1998]). The plots indicate the contributions of process variables to an observed increase in the T^2 or Q statistic, with variables showing large contributions concluded as no longer following nominal operating conditions. Operators can then apply process knowledge to determine an appropriate cause.

There are reports of fault contribution plots suffering from a property called "fault smearing" - variables unaffected by the fault demonstrate a contribution and actual faulty variables are obscured (Westerhuis et al. [2000]). Smearing occurs because the compression of the input to a smaller number of latent variables and subsequent expansion to the original space permits faulty and non-faulty variables to interact (Van den Kerkhof et al. [2013]). Gao et al. [2016] imposed an elastic net constraint to obtain a sparse PCA model for the Tennessee Eastman process. The result was a reduction in interactivity between variables in the latent space. It subsequently lead to the discovery of process knowledge, specifically the relationships among process variables.

The objective of this paper is to extend the analysis of fault contribution plots to AEs and investigate the effect reduced latent variable interactivity has on process variable contribution. Two AEs - a dense one and a sparse one - are generated to monitor a numerical simulation of a nonlinear triple tank process (TTP) - a variant of the quadruple tank process (QTP) (Johansson [2000]). Their ability to detect a fault is demonstrated by inducing an abnormal bias in one of the TTP's sensors. Individual contributions of process variables to the AEs' monitoring statistics are then analyzed. The key result in this study was that sparsity reduced fault smearing onto non-faulty variables and increased the contributions of faulty variables.

This paper presents the mathematical model of the TTP in section II. Section III describes how a sparse AE is obtained and subsequently used in process monitoring. The effectiveness of the sparse AE method at process monitoring and improved generation of fault contribution plots is presented in section IV.

2. THE TRIPLE TANK PROCESS

A schematic drawing of the TTP is given in Fig. 1. The upper tanks are supplied with liquid that is transported from a large sump by the means of two gear pumps. Liquid flows from the upper left tank into the sump. The liquid from the upper right tank flows into the lower tank, which sequentially flows into the sump. The objective is to control the liquid levels in the upper left and lower right tanks, which are monitored with two voltage-based level measurement devices. A level measurement device is also fixed to the upper right tank. A nonlinear numerical model of the TTP is derived by applying mass balances and Bernouilli's law to yield a set of differential equations that describes the evolution of the liquid level of each tank. They are:

$$\begin{aligned}\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{1}{2}\frac{k_1}{A_1}v_1(1+\eta_1)\\ \frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{1}{2}\frac{k_1}{A_2}v_1(1+\eta_1) + \frac{k_2}{A_2}v_2(1+\eta_2)\\ \frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{a_2}{A_3}\sqrt{2gh_2}\end{aligned} \tag{1}$$

where A_i is the cross-section of tank i and a_i is the cross-section of its outlet hole. The liquid level of tank i is h_i and g is acceleration due to gravity. The voltage applied to pump i is v_i and the corresponding flow is $k_i v_i(1+\eta_i)$, where $\eta_i \in \mathbb{R}$ is zero mean Gaussian noise emitted from pump i. The system is measured and actuated discretely with a sample time of T_s. The measured level signals at sample k are:

$$\begin{aligned}y_1[k] &= k_c h_1[k] + w_1[k]\\ y_2[k] &= k_c h_2[k] + w_2[k]\\ y_3[k] &= k_c h_3[k] + w_3[k]\end{aligned} \tag{2}$$

where $w_i[k] \in \mathbb{R}$ is zero mean measurement noise with Gaussian distribution for level signal i. For decentralized control, the error terms are:

$$\begin{aligned} e_1[k] &= r_1[k] - y_1[k] \\ e_2[k] &= r_2[k] - y_3[k] \end{aligned} \tag{3}$$

where $r_1[k]$ and $r_2[k]$ are reference signals for level signals $y_1[k]$ and $y_3[k]$, respectively. The error terms are minimized by a discrete PI controller. The closed loop control laws for the process inputs are:

$$\begin{aligned} K_1 :\ v_1[k] &= K_P e_1[k] + K_I \sum_{i=1}^{k} e_1[i] T_s \\ K_2 :\ v_2[k] &= K_P e_2[k] + K_I \sum_{i=1}^{k} e_2[i] T_s \end{aligned} \tag{4}$$

Here K_P and K_I denote the proportional and integral gains, respectively, of the PI controller. Monte Carlo simulations were performed on the TTP to generate data sets that exhibited nonlinear correlations between the process variables. The data sets were used to train, validate, and test an AE model that monitored the process. The uncertain parameters were the reference signals r_1 and r_2. Values for r_1 and r_2 were sampled from two independent uniform distributions. Process, controller, and noise parameters were based on the QTP from Johansson [2000] and are listed in Table 1.

3. AUTOENCODERS

Process variables tend to be highly correlated with one another due to the presence of physical laws and control loops in process plants. Feature extraction can be performed on the original variable space to reveal the simplified structure that underlies it. An AE - an artificial neural network used for learning encoded representations for a set of data - is applicable when variables exhibit nonlinear correlations. Given a $m \times 1$ vector of process variables $\mathbf{x}$, the $m \times n$ reference data matrix consisting of n standardized observations is:

$$\mathbf{X} = \begin{array}{c} \begin{matrix} \mathbf{x}[1] & \mathbf{x}[2] & \cdots & \mathbf{x}[n] \end{matrix} \\ \begin{bmatrix} x_1[1] & x_1[2] & \cdots & x_1[n] \\ x_2[1] & x_2[2] & \cdots & x_2[n] \\ \vdots & \vdots & \ddots & \vdots \\ x_m[1] & x_m[2] & \cdots & x_m[n] \end{bmatrix} \end{array} \in \mathbb{R}^{m \times n} \tag{5}$$

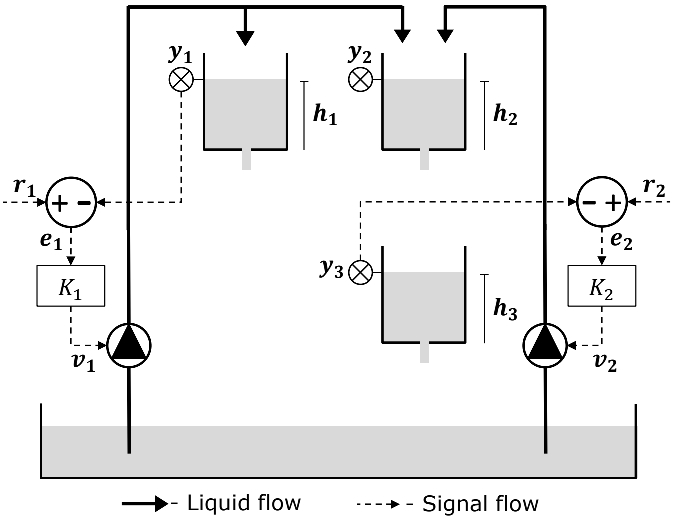

Fig. 1. A schematic of the TTP showing the connectivity of the tanks and location of the pumps, dual valves, and the level measurement devices. Included are the decentralized feedback loops.

Table 1. List of Parameters

Process param.		**Noise param.**	
A_1, A_3	28 cm^2	η_i	$\mathcal{N}(0, 0.1)$
A_2	32 cm^2	w_i	$\mathcal{N}(0, 0.0005)$
a_1, a_3	0.071 cm^2		
a_2	0.057 cm^2		
k_c	1 V/cm	**Controller param.**	
k_1	3.33 cm^3/Vs	T_s	10
k_2	3.35 cm^3/Vs	K_P	20
g	981 cm/s^2	K_I	0.25

An AE consists of two parts - an encoder and a decoder. The encoder transforms its input $\mathbf{X}$ into new, higher-level representative features $\mathbf{Z} \in \mathbb{R}^{q \times n}$. The decoder then reconstructs the original data as $\hat{\mathbf{X}} \in \mathbb{R}^{m \times n}$ with a transformation of the features (Hinton [2006]). Modifiable interconnecting weights are introduced in the AE such that it learns in an unsupervised manner to minimize the difference between its input and its reconstruction.

The simplest form of an AE is a multilayered, feedforward, non-recurrent neural network. Nonlinear transformations occur at the layers of the network, allowing for processing of data that has inherent nonlinear properties. The encoder maps the input $\mathbf{X} \in \mathbb{R}^{m \times n}$ to the latent variables $\mathbf{Z} \in \mathbb{R}^{q \times n}$:

$$\begin{aligned} \mathbf{E}_i &= \begin{cases} \sigma_1^e \left(\mathbf{W}_1^e \mathbf{X} + \mathbf{b}_1^e \right), & \text{for } i = 1 \\ \sigma_i^e \left(\mathbf{W}_i^e \mathbf{E}_{i-1} + \mathbf{b}_i^e \right), & \text{else} \end{cases} \\ \mathbf{Z} &= \sigma^z \left(\mathbf{W}^z \mathbf{E}_N + \mathbf{b}^z \right) \end{aligned} \tag{6}$$

where $i \in \mathbb{Z} : i \in [1, N]$. $\mathbf{W}_1^e$ is the weight matrix between the input layer and the first encoder layer. $\mathbf{W}_i^e$ is the weight matrix between layers $i-1$ and i, $\mathbf{b}^e$ is the bias at layer i, and σ_i^e is the component wise activation function at layer i. $\mathbf{W}^z$, $\mathbf{b}^z$, and σ^z are defined similarly for the latent layer.

The decoder maps the latent variables $\mathbf{Z} \in \mathbb{R}^{q \times n}$ to the input reconstruction $\hat{\mathbf{X}} \in \mathbb{R}^{m \times n}$:

$$\begin{aligned} \mathbf{D}_i &= \begin{cases} \sigma_1^d \left(\mathbf{W}_1^d \mathbf{Z} + \mathbf{b}_1^d \right), & \text{for } j = 1 \\ \sigma_j^d \left(\mathbf{W}_j^d \mathbf{D}_{j-1} + \mathbf{b}_j^d \right), & \text{else} \end{cases} \\ \hat{\mathbf{X}} &= \sigma^{\hat{x}} \left(\mathbf{W}^{\hat{x}} \mathbf{D}_M + \mathbf{b}^{\hat{x}} \right) \end{aligned} \tag{7}$$

where $j \in \mathbb{Z} : j \in [1, M]$. $\mathbf{W}_1^d$ is the weight matrix between the latent layer and the first decoder layer. $\mathbf{W}_j^d$ is the weight matrix between layers $j-1$ and j, $\mathbf{b}^d$ is the bias at layer j, and σ_j^d is the component wise activation function at layer j. $\mathbf{W}^{\hat{x}}$, $\mathbf{b}^{\hat{x}}$, and $\sigma^{\hat{x}}$ are defined similarly for the output layer. The modifiable parameters $\mathbf{W}_i^e, \mathbf{b}_i^e, \mathbf{W}^z, \mathbf{b}^z, \mathbf{W}_j^d, \mathbf{b}_j^d, \mathbf{W}^{\hat{x}}$, and $\mathbf{b}^{\hat{x}}$ are optimized with respect to minimizing the following reconstruction loss function via stochastic gradient descent (Nielsen [2015]):

$$\mathcal{L}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{n} \left| \left| \mathbf{X} - \hat{\mathbf{X}} \right| \right|^2 \tag{8}$$

Fig. 2 illustrates a typical autoencoder that gradually condenses the input to the latent space and then gradually reconstructs it. The dimension q of the latent layer plays a significant role in the discovery of informative representations of the input. The traditional approach is to create a bottleneck by setting $q < m$, thereby forming an under-complete representation. In this case, the network pursues an effective compression that retains information about input $\mathbf{X}$. The compressed data, being sufficiently representative of the original data, allows for accurate reconstruction of the input data, albeit with a non-zero reconstruction error. An AE using linear nodal activation functions will uncover latent projection that correspond to the projection onto the subspace obtained from PCA of the covariance matrix of $\mathbf{X}$ (Baldi and Hornik [1989]). This occurs even if the network is composed of several layers of linear units. However, Bourlard and Kamp [1988] show that PCA-like projections can be obtained even if nonlinear functions are used since it is possible for activations to remain in the linear regions of functions such as the sigmoid or tangent hyperbolic. This becomes unlikely if the AE is composed of several hidden layers with varying activation functions (Japkowicz et al. [2000]).

3.1 Invoking network sparsity

Further optimization constraints are introduced to obtain latent representations that generalize better and prevent over-fitting. One approach is to include the naïve elastic net weight decay penalty - a regularized regression method that linearly combines the L_1 and L_2 weight decay penalties of the LASSO and ridge methods (Zou and Hastie [2005]). The loss function in (8) becomes:

$$\mathcal{L}(\mathbf{X}, \hat{\mathbf{X}}, \mathbf{W}) = \frac{1}{n}\left|\left|\mathbf{X} - \hat{\mathbf{X}}\right|\right|^2 + \lambda_1 \left|\left|\mathbf{W}\right|\right|_1 + \lambda_2 \left|\left|\mathbf{W}\right|\right|_2^2 \quad (9)$$

where λ_1 and λ_2 control the importance of the LASSO and ridge regressions, respectively, and $\mathbf{W}$ is the collection of weight matrices in (6) and (7). Biases $\mathbf{b}$ in (6) and (7) are not included in the naïve elastic net penalty. Minimization of (9) yields an optimized AE consisting of shrunk weights that minimize its reconstruction loss. The individual contribution of each regularization term is: (a) L_1 regularization shrinks weights at a constant rate towards zero, thereby establishing a small number of high-magnitude, i.e., high-importance, connections by driving redundant weights to zero; and (b) L_2 regularization shrinks weights by an amount proportional to their magnitude, thus penalizing larger weights more than smaller weights. The net result is an interpretable grouping of correlated variables; L_2 regularization opposes the tendency of L_1 regularization to prioritize one variable from a group correlated variables and ignore the others. Grouping of process variables is relevant for identification of control systems; Gao et al. [2016] demonstrate that a sparse principal component model can uncover the underlying process variable relations.

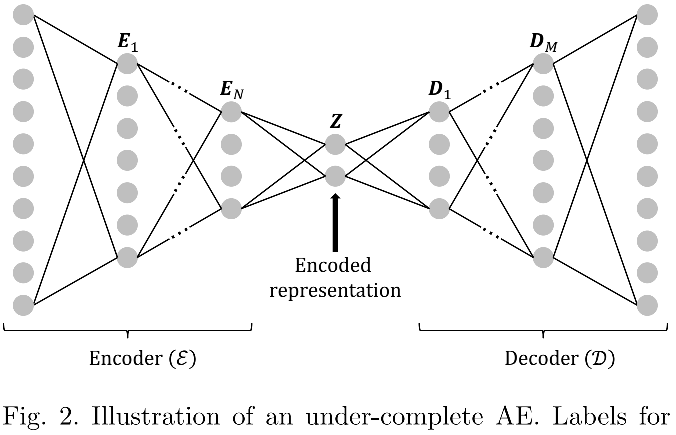

Fig. 2. Illustration of an under-complete AE. Labels for the encoder and decoder of the network are included. Biases are excluded from the illustration.

Weight connections deemed redundant can be removed to clarify the interconnectivity of a neural network. Magnitude-based weight pruning is a technique that reduces the number of non-zero weight parameters to invoke network sparsity. Zhu and Gupta [2017] introduce a pruning algorithm that progressively trims away redundant weight connections. Weight connections are removed according to a pruning function that sets the current sparsity percentage, i.e., the ratio of the number zero magnitude weights to the total number of weights, of a network:

$$s_t = s_f + (s_i - s_f)\left(1 - \frac{t - t_0}{n\Delta t}\right)^3$$
$$\text{for } t \in \{t_0, t_0 + \Delta t, \ldots, t_0 + n\Delta t\} \quad (10)$$

The network is first trained for t_0 time steps to permit the weights to converge to an acceptable solution. Thereafter the initial sparsity of the network is set to s_i (usually zero). Weights are then pruned every Δt steps to gradually increase the network's sparsity while allowing it to recover from any pruning-induced loss in accuracy. The intuition behind the order of (10) is to rapidly prune the network in the beginning phase when redundant connections are plentiful before slowing down once fewer connections remain (Fig. 3). The algorithm operates continuously over n sparsity updates until the final sparsity value s_f is reached. Zhu and Gupta [2017] discovered that large-sparse models consistently outperformed small-dense models when the number of parameters was kept the same.

The pruning algorithm presented by Zhu and Gupta [2017] is extended upon in this paper. At every sparsity update s_t, each weight matrix $\mathbf{W}_i \in \mathbf{W}$ is divided by the largest absolute value of $\mathbf{W}_i$. This normalization step is done to prevent severe pruning of weight matrices whose largest absolute value is much smaller compared to the other matrices. The normalized matrices are then flattened and concatenated. The smallest weights are then masked to zero until the desired sparsity level s_t is reached. Furthermore, the pruning algorithm is stopped prematurely if the validation loss experiences a 5%-10% increase.

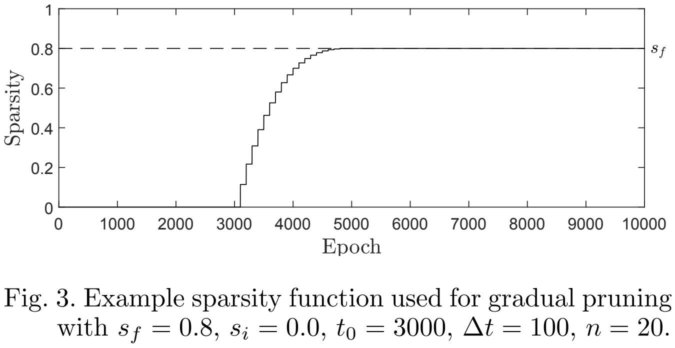

Fig. 3. Example sparsity function used for gradual pruning with $s_f = 0.8$, $s_i = 0.0$, $t_0 = 3000$, $\Delta t = 100$, $n = 20$.

Once pruning ends, the näive elastic net weight penalty is removed from the training session. This is to relax the constraints on the AE and permit the remaining weights to maximize their capacity to reduce the loss function in (8) without the concern of any additional loss penalties.

3.2 Process Monitoring

Process monitoring consists of comparing current plant behaviour with that predicted by an "in-control" AE trained with historical data collected when the process exhibited nominal behaviour. New observations are propagated through the AE to generate the residuals $\mathbf{e}_{new} = \mathbf{x}_{new} - \hat{\mathbf{x}}_{new}$. The quality of new observations is assessed by computing the squared prediction error (SPE) (more formally known as the Q statistic) of the residuals of new observations (MacGregor et al. [1995]):

$$SPE = \sum_{i=1}^{m} (x_{new,i} - \hat{x}_{new,i})^2 \tag{11}$$

An abnormal event that changes the correlation between process variables will cause the SPE to increase. Assuming that the SPE follows a chi-squared distribution, the control limit can be computed with the following approximate value (Box [1954]):

$$CL_{SPE_{AE}} = \frac{\bar{\sigma}^2}{2\bar{\mu}}\chi^2_{(2\bar{\mu}^2/\bar{\sigma}^2,\alpha)} \tag{12}$$

where $\bar{\mu}$ and $\bar{\sigma}$, respectively, are the sample mean and sample standard deviation of the SPE and α is the false alarm rate. An abnormal event is deemed to have occurred if the SPE crosses the control limit. Abnormal process variables are isolated by analysing the contribution of each variable i to the SPE in (11) (Miller et al. [1998]):

$$C_i = (x_{new,i} - \hat{x}_{new,i})^2 \tag{13}$$

Variables with large contributions are said to no longer be consistent with normal operating conditions. It is noted that analysis of (13) does not determine the underlying cause of a fault. Rather, it will highlight the process variables containing signal characteristics of a fault. The results of (13) must be integrated with a qualitative model of the process that takes into account the causal nature of system components to decipher the actual cause.

4. RESULTS AND DISCUSSION

4.1 Derivation of influence rules

It was of practical interest to determine the influence of reference variables r_1 and r_2 on the control and measurement variables; steady-state correlations uncovered by the AE can then be validated to what is implied by the data. Fig. 4 displays the time series plots obtained from inducing random step changes in a single reference variable while keeping the other constant. The plots demonstrate that: (a) a step change in r_1 causes a transient change in the steady state values of y_1, v_1, and v_2, while variables y_2 and y_3 experience a transient change that has no affect on their steady-state values; and (b) a step change in r_2 has no influence on y_1 and v_1 yet generates a transient change in the steady state values of y_2, y_3 and v_2. The correlation sets $C_1 = (r_1, y_1, v_1, v_2)$ and $C_2 = (r_2, y_2, y_3, v_2)$ are determined from the plots. They indicate which process variables observe a permanent change in their steady state value caused by a change in a reference signal.

4.2 Data generation from TTP simulation

The TTP was simulated with random step changes in reference signals r_1 and r_2 occurring every 200 time steps. The training set $\mathbf{X}_t$ (consisting of 300,000 samples) and validation set $\mathbf{X}_v$ (consisting of 30,000 samples) were generated to train and validate, respectively, an AE. Fig. 5 displays the distribution of standardized samples of variables in $\mathbf{X}_t$ in the form of scatter and histogram plots. The scatter plots indicate the existence of nonlinear correlations between variable pairs (r_1, v_1), (v_1, y_1), and (v_2, y_1). The histogram plots reveal that several variables do not follow the assumption of normality with v_1 in particular.

The fault set $\mathbf{X}_f$ (consisting of 300 samples) was generated by simulating the TTP with a bias in sensor 1, introduced with the additive fault $y_1[k] = k_c h_1[k] + w_1[k] + f$ with $f = -0.01$. The fault was introduced after 100 time steps. No reference changes occurred in r_1 and r_2. Fig. 6 presents time series plots of the first 200 samples of $\mathbf{X}_f$ and shows the fault's effect on the process variables. Deterministic results (in grey) from the same simulation case (obtained by setting η_i and w_i in (1), (2) to zero) are included to aid in interoperability. The plots demonstrate that: (a) the fault has no influence on r_1 and r_2; (b) the fault induces temporary changes in y_1, y_2, and y_3 that have no influence on their steady state values; and (c) the fault induces a permanent change in v_1 and v_2 and thus carry steady-state signatures that explain the presence of the fault.

4.3 AE model generation and testing

Two AEs, denoted AE_1 and AE_2, were trained with the training set $\mathbf{X}_t$. Both networks were inherently the same,

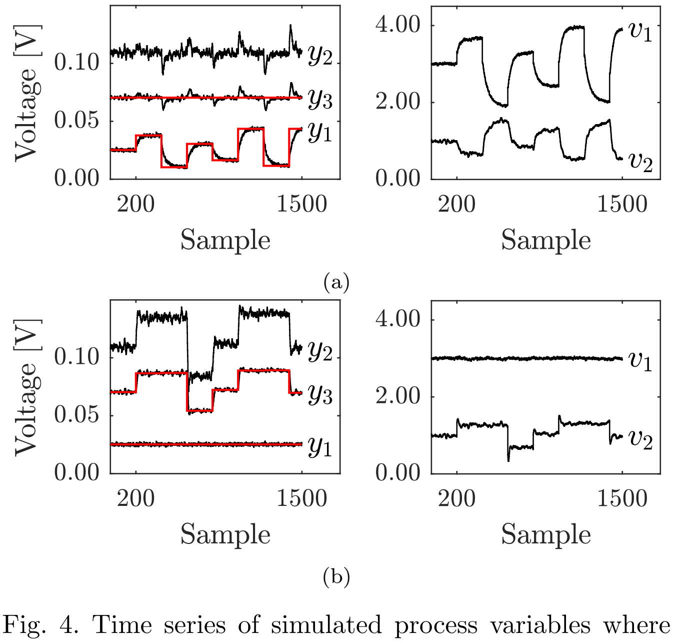

Fig. 4. Time series of simulated process variables where (a) r_1 is changed whilst r_2 is held fixed and (b) r_2 is changed whilst r_1 is held fixed. Red lines indicate the references for the measurement.

i.e., same number and dimension of layers, same number of latent variables, same initialization of the weights, and so on, except AE_2 included the näive elastic net weight penalty with $\lambda_1 = \lambda_2 = 0.001$ and was pruned. The pruning parameters in (10) of AE_2 were $s_f = 0.9$, $s_i = 0.7$, $t_0 = 5000$, $\Delta t = 100$, $n = 200$, but early-stopping resulted with a sparsity of 80.86%. Both AEs were trained for 12000 epochs using the Adam gradient-based optimization with a learning rate of 0.001 for stochastic gradient descent (Kingma and Ba [2014]). The matrices $\mathbf{X}_t$, $\mathbf{X}_v$, and $\mathbf{X}_f$ were standardized with the mean and standard deviation of $\mathbf{X}_t$. The dimension of the latent layer in each model was set to $q = 2$. This was to see if the sparse AE_2 would expose the correlation sets C_1 and C_2. The dimensions and activation function of each layer were specified as:

$$\left[\begin{array}{c|c}\dim_{\mathrm{L}}(\mathcal{E}) & \dim_{\mathrm{L}}(\mathcal{D}) \\ \sigma_i^e & \sigma_i^d\end{array}\right] = \begin{array}{ccc|ccc} \mathbf{X} & \mathbf{E}_1 & \mathbf{E}_2 & \mathbf{D}_1 & \mathbf{D}_2 & \hat{\mathbf{X}} \\ 7 & 9 & 9 & 9 & 9 & 7 \\ & \tanh & \tanh & \tanh & \tanh & I \end{array} \tag{14}$$

where tanh is the tangent hyperbolic function and I is the identity function. The tangent hyperbolic transfer function was primarily used since the data is mean-centered. The design of the AE is essentially an expanded under-represented AE; setting the dimension of the encoder and decoder layers larger than the size of the input dimension allowed the AE models to generate complex, higher dimensional features before information retaining compression occurred (Olah [2014]). The tangent hyperbolic function was implemented at the latent layer.

The training loss (TL) and validation loss (VL) from training AE_1 and AE_2 are plotted in Fig. 7. It can be seen that the TL and VL of AE_2 observe a significant difference that recedes when pruning ends. This is because the TL includes the näive elastic net weight penalty in (9) that is then removed once pruning stops. The figure shows that the VL of AE_2 is similar to the VL of AE_1 at the end of training. In fact, the VL of AE_2 is only 7.2% larger despite AE_2 having 80.86% fewer weight parameters than AE_1.

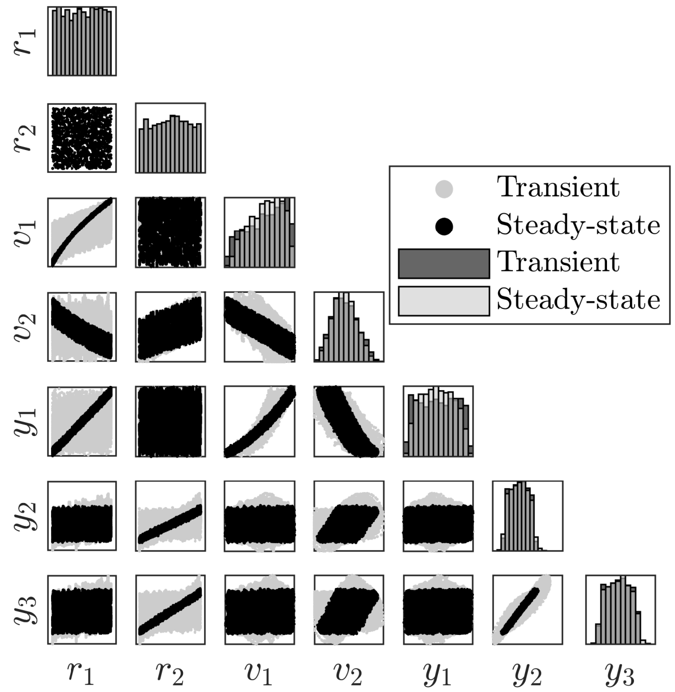

Fig. 5. Scatter plot of standardized process variables, including a histogram along the diagonal.

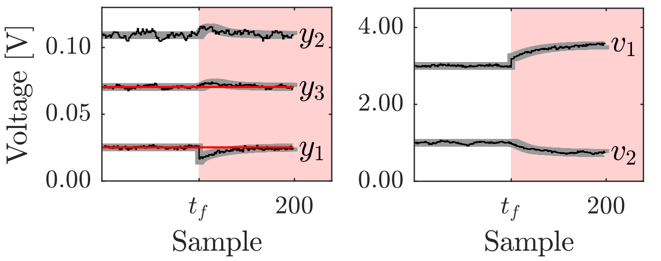

Fig. 6. Influence of fault f_1 at sample t_f on (left) measurements and (right) control inputs. Red lines indicate the references for the measurements.

Fig. 8 portrays the connectivity between network layers of AE_2 and shows the propagation of original variables $\mathbf{x}$ to the reconstructions $\hat{\mathbf{x}}$. It can be seen that the network has identified the correlations between process variables, thus eliminating potential fault smearing between uncorrelated variables. The activation of the second node in the latent layer is computed by the process variables of correlation set C_1. In addition, the activation is solely responsible for the reconstruction of the same variables. The activation of the first latent node is determined by the variables of correlation set C_2 with the exception of v_2. However, the latent node's activation reconstructs all of the variables of C_2. From this it follows that fault signatures contained in v_2 cannot not smear onto $\hat{r}_2$, $\hat{y}_2$, and $\hat{y}_3$. Although it provides a partial explanation for the loss in validation accuracy in comparison to AE_1 (Fig. 7), AE_2 has discovered a form of reconstruction redundancy: although v_2 appears both in C_1 and C_2, it is sufficient to reconstruct it from a partial subset of process variables. It is noted that the interconnectivity of AE_2 is heavily influenced by the chosen hyperparameters for the learning rate, regression coefficients λ_1 and λ_2, and pruning parameters; a different selection is bound to result with a different connectivity.

The contribution plots obtained from propagating $\mathbf{X}_f$ through AE_1 and AE_2 are displayed in Fig. 9. Plots from the deterministic equivalent of $\mathbf{X}_f$ are included to ease the analysis of the effect of network pruning on mean contributions. The fault is detected by both AEs as their SPEs cross their control limit at sample t_f, i.e., the onset of the fault. Despite the model complexity of AE_1 being greater than that of AE_2, their SPEs are nearly identical over the fault set. This indicates that a more

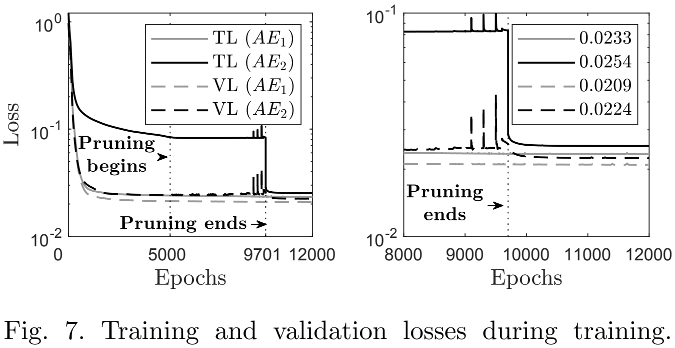

Fig. 7. Training and validation losses during training. Right figure zooms in on epoch interval [8000,12000] and includes final losses in its legend.

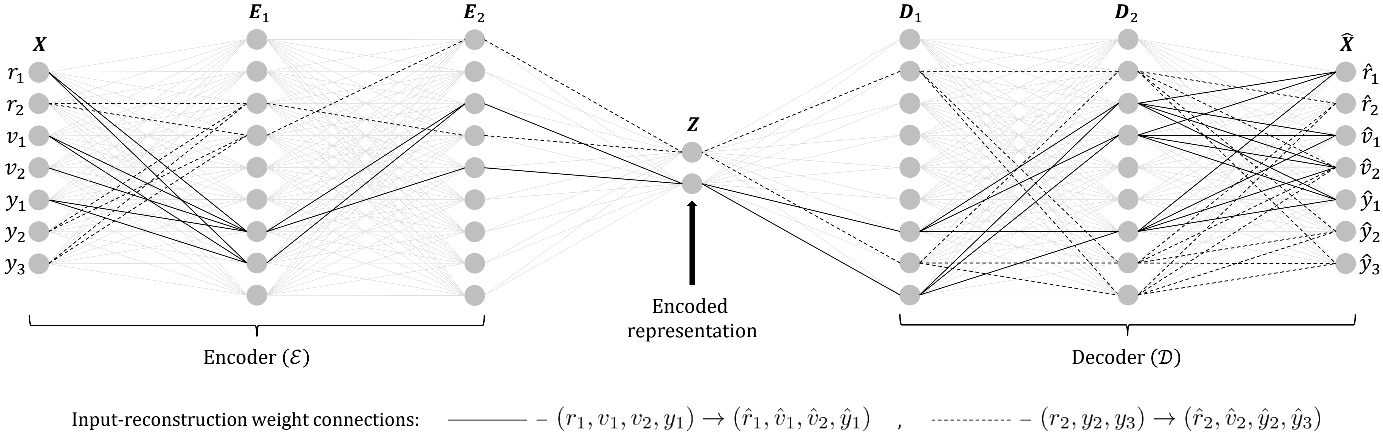
X
E₁
E₂
Z
D₁
D₂
X̂
r₁
r₂
v₁
v₂
y₁
y₂
y₃
r̂₁
r̂₂
v̂₁
v̂₂
ŷ₁
ŷ₂
ŷ₃
Encoded representation
Encoder (ℰ)
Decoder (𝒟)
Input-reconstruction weight connections:
– (r₁, v₁, v₂, y₁) → (r̂₁, v̂₁, v̂₂, ŷ₁)
– (r₂, y₂, y₃) → (r̂₂, v̂₂, ŷ₂, ŷ₃)

Fig. 8. Illustration of trained AE_2, showing pruned weight connections (grey) and remaining connections (black). Biases have been excluded from the illustration.

complex model is not necessarily more sensitive to faults. Smearing onto unaffected variables r_2, y_2, and y_3 is less for AE_2, indicated by a reduction in the variance (Fig 9a) and mean (Fig 9b) of their contributions. In fact, their mean contributions are zero once steady-state is reached because the steady state fault signatures retained in v_1 and v_2 cannot propagate to $\hat{r}_2$, $\hat{y}_2$, and $\hat{y}_3$ (Fig. 8). Even though smearing occurs onto non fault-carrying variables r_1 and y_1, invoking network sparsity guarantees that the steady state signal characteristics of faulty variables v_1 and v_2 stay within the variables of correlation set C_1. In fact, network AE_2 generates larger contributions for fault-carrying variables v_1 and v_2 and reduces the contributions for non-fault-carrying variables r_1 and y_1, indicating that network sparsity makes faulty variables more highlighted.

It is reiterated that analysis of fault contribution plots does not determine the cause of a fault. Instead, process variables containing steady-state fault signatures are inferred. An additional “causal reasoning” step must be performed that takes into consideration the causal nature of the monitored process, e.g, qualitative modeling of relations between different components of a system, to determine the root cause of fault-contaminated process variables. The presented method makes qualitative diagnosis more effective, since the reduction of fault smearing ensures that more precise qualitative information is provided.

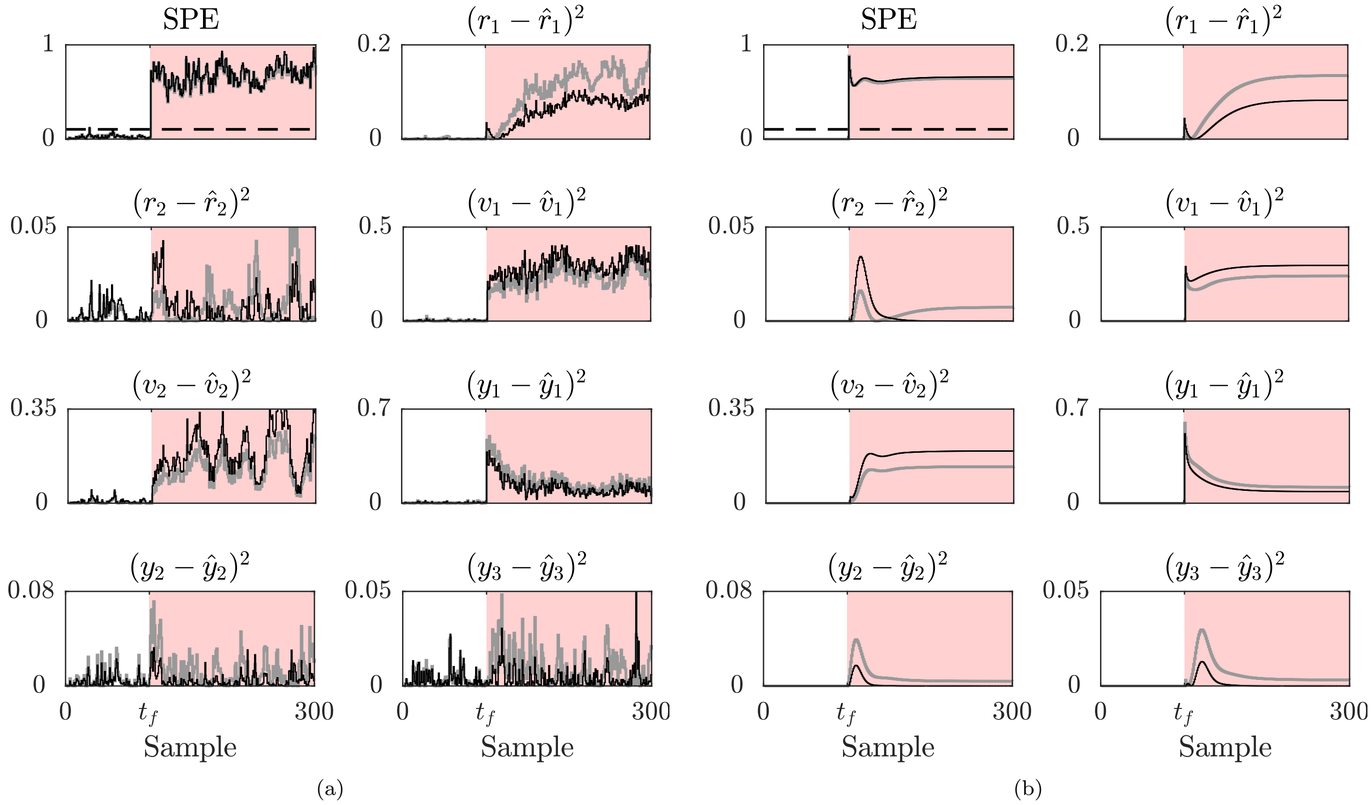

Fig. 9. Magnitudes of contributions to the SPE from AE_1 (grey) and AE_2 (black) via (a) stochastic simulations for $\mathbf{X}_f$ and (b) deterministic simulations for $\mathbf{X}_f$. Dashed line in SPE plot indicates the control limit obtained from $\mathbf{X}_v$.

5. CONCLUSION

This study introduces the combined application of a sparsity constraint and a pruning strategy to produce a sparse AE with the purpose of diagnosing a sensor fault occurring in the TTP. The obtained AE lead to the discovery of process knowledge, specifically the relationships among process variables. The solution demonstrated that a sparse AE, which inherently has fewer parameters than a fully connected AE, suffered little in its validation performance.

The results show that the proposed method improved the performance of fault contribution plots; process variables unaffected by the fault produced significant less contributions due a reduction of fault smearing. The results also demonstrated that variables carrying no fault signatures, but were strongly correlated with the faulty variables, observed reduced contributions. Finally, variables that contained fault signatures produced larger contributions, providing further fault isolation capabilities.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the Danish Hydrocarbon Research and Technology Center (DHRTC) at the Technical University of Denmark.

REFERENCES

P. Baldi and K. Hornik. Neural networks and principal component analysis: learning from examples without local minima. *Neural Networks, 2(1):53-58,1989.*

H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics, 59(4-5): 291-294, 1988.*

G.E. Box. Some theorems on quadratic forms applied in the study of analysis of variance problems, I. Effect of inequality of variance in the one-way classification. *The Annals of Mathematical Statistics, 25(2):290-302, 1954.*

E.B. Brooks et al. On-the-fly massively multitemporal change detection using statistical quality control charts and Landsat data. *IEEE Transactions on Geoscience and Remote Sensing, 52(6):3316-3332, 2014.*

P. Bullemer et al. ASM consortium guidelines-effective operator display design. *Houston, Honeywell International Inc./ASM Consortium, 2008.*

F. Cheng, Q.P. He, and J. Zhao. A novel process monitoring approach based on variational recurrent autoencoder. *Computers & Chemical Engineering, 129:106515, 2019.*

D. Dong and T.J. McAvoy. Nonlinear principal component analysis-based on principal curves and neural networks. *Computers & Chemical Engineering 20(1):65-78, 1996.*

H. Gao et al. Process knowledge discovery using sparse principal component analysis. *Industrial & Engineering Chemistry Research, 55(46):12046-12059, 2016.*

Á.D. Hallgrímsson, H.H. Niemann, and M.Lind. Autoencoder based residual generation for fault detection of quadruple tank system. *IEEE Conference on Control Technology and Applications (CCTA), p:994-999, 2019.*

G.E. Hinton. Reducing the dimensionality of data with neural networks. *Science, 313(5786):504-507, 2006.*

N. Japkowicz, S.J. Hanson, and M.A. Gluck. Nonlinear autoassociation is not equivalent to PCA. *Neural Computation, 12(3): 531-545, 2000.*

S. Joe Qin. Statistical process monitoring: basics and beyond. *Journal of Chemometrics: A Journal of the Chemometrics Society 17(8-9):480-502. 2003.*

K.H. Johansson. The quadruple tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on Control System Technology, 8(3):456-465, 2000.*

D.P. Kingma and J. Ba Adam: A method for stochastic optimization. *arXiv preprint, arXiv:1412.6980, 2014.*

M.A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal, 37(2): 233-243, 1991.*

S. Lee et al. Process monitoring using variational autoencoder for high-dimensional nonlinear processes. *Engineering Applications of Artificial Intelligence 83:13-27, 2019.*

J.F. MacGregor and T. Kourti. Statistical process control of multivariate processes. *Control Engineering Practice, 3(3):403-414, 1995.*

P. Miller, R.E. Swanson, and C.E. Heckler. Contribution plots: a missing link in multivariate quality control. *Applied Mathematics and Computer Science, 8(4):775-792, 1998.*

M.A. Nielsen. Neural networks and deep learning. *Vol. 2018. USA: Determination Press, 2015.*

C. Olah. Neural networks, manifolds, and topology. url: *http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/, 2014.*

A. Osmani, M. Hamidi, and S. Bouhouche. Monitoring of a dynamic system based on autoencoders. *Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press:1836-1843, 2019.*

F.A.P. Peres and F.S. Fogliatto. Variable selection methods in multivariate statistical process control: a systematic literature review. *Computers & Industrial Engineering, 115:603-619, 2018.*

W. Sun et al. A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement 89: 171-178, 2016.*

P. Van den Kerkhof et al. Contribution plots for statistical process control: Analysis of the smearing-out effect. *In 2013 European Control Conference (ECC). IEEE:428-433, 2013.*

J.A. Westerhuis, S.P. Gurden, and A.K. Smilde. Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems, 51(1):95-114, 2000.*

J. Yan et al. Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance. *IEEE Access 5, 23484-23491, 2017.*

W. Yan, P. Guo, and Z. Li. Nonlinear and robust statistical process monitoring based on variant autoencoders. *Chemometrics and Intelligent Laboratory Systems 158:31-40, 2016.*

M. Zhu and S. Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878, 2017.*

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2):301-320, 2005.*