Tensor-Based Statistical Learning Methods for Diagnosing Product Quality Defects in Multistage Manufacturing Processes

Cheoljoon Jeong[a], Eunshin Byon[a], Fei He[b], and Xiaolei Fang[c]

[a]Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, USA
[b]The Collaborative Innovation Center of Steel Technology, University of Science and Technology Beijing, Beijing 100083, China
[c]Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, NC 27695, USA

Abstract

Multistage manufacturing processes with identical stages provide three-dimensional process data in which the first dimension represents the process (control/sensing) variable, the second is the stage, and the third is the measurement/sampling/data acquisition time point. Diagnosing quality faults in such processes often requires the simultaneous identification of crucial process variables and stages associated with product quality anomalies. Most existing diagnosis methods convert 3D data into a 2D matrix, resulting in loss of information and reduced diagnostic accuracy and stability. To address this challenge, we propose a penalized tensor regression model that regresses the product quality index against its 3D process data. For the estimation of high-dimensional regression coefficients with the limited amount of historical data, we apply the CANDECOMP/PARAFAC and Tucker decompositions to the coefficient tensor, which significantly reduces the number of parameters to be estimated. Based on the decompositions, a new regularization term is designed to enable the joint identification of critical process variables and stages. To estimate the parameters, we develop the block coordinate proximal descent algorithm and provide its convergence guarantee. Numerical studies demonstrate that the proposed methods can enhance diagnostic stability and on average improve the diagnostic accuracy by around 20% over existing benchmarks.

Keywords: Quality defect/fault diagnosis; Penalized tensor regression; Two-dimensional variable selection.

Nomenclature

Dimensions & Ranks

I_d, P_d Dimension of dth mode of a tensor, $d = 1, 2, 3$.

R_d, R Rank of dth mode of a tensor, $d = 1, 2, 3$.

Variables

$\boldsymbol{x}$ A vector in $\mathbb{R}^{I_1}$ or one-dimensional tensor denoted by a boldface lowercase letter.

$\boldsymbol{X}$ A matrix in $\mathbb{R}^{I_1 \times I_2}$ or two-dimensional tensor denoted by a boldface uppercase letter.

$\mathcal{X}$ A tensor in $\mathbb{R}^{I_1 \times I_2 \times I_3}$ or a three-dimensional tensor denoted by a calligraphic letter.

$x_{i_1 i_2 i_3}$ An $(i_1, i_2, i_3)th$ entry in $\mathbb{R}$ of a tensor $\mathcal{X}$, $i_1 = 1, \ldots, I_1, i_2 = 1, \ldots, I_2, i_3 = 1, \ldots, I_3$.

$\boldsymbol{x}_{:i_2 i_3}$ A mode-1 (column) fiber in $\mathbb{R}^{I_1}$ of a tensor $\mathcal{X}$, $\forall i_2, i_3$.

$\boldsymbol{x}_{i_1 : i_3}$ A mode-2 (row) fiber in $\mathbb{R}^{I_2}$ of a tensor $\mathcal{X}$, $\forall i_1, i_3$.

$\boldsymbol{x}_{i_1 i_2 :}$ A mode-3 (tube) fiber in $\mathbb{R}^{I_3}$ of a tensor $\mathcal{X}$, $\forall i_1, i_2$.

$\boldsymbol{X}_{i_1 ::}$ A mode-1 (horizontal) slice in $\mathbb{R}^{I_2 \times I_3}$ of a tensor $\mathcal{X}$, $\forall i_1$.

$\boldsymbol{X}_{:i_2 :}$ A mode-2 (lateral) slice in $\mathbb{R}^{I_1 \times I_3}$ of a tensor $\mathcal{X}$, $\forall i_2$.

$\boldsymbol{X}_{::i_3}$ A mode-3 (frontal) slice in $\mathbb{R}^{I_1 \times I_2}$ of a tensor $\mathcal{X}$, $\forall i_3$.

Operators

$\text{vec}(\mathcal{X})$ A vectorization of a tensor $\mathcal{X}$, which stacks all mode-1 fibers of a tensor $\mathcal{X}$ into one vector.

$\langle \mathcal{X}, \mathcal{Y} \rangle$ An inner product of two same-sized tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, which calculates $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1, i_2, i_3} x_{i_1 i_2 i_3} y_{i_1 i_2 i_3}$.

$\boldsymbol{X} \otimes \boldsymbol{Y}$ A Kronecker product of two matrices $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{m \times n}$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_q] \in \mathbb{R}^{p \times q}$, which is defined by $\boldsymbol{X} \otimes \boldsymbol{Y} = [\boldsymbol{x}_1 \otimes \boldsymbol{Y}, \ldots, \boldsymbol{x}_n \otimes \boldsymbol{Y}] = [\boldsymbol{x}_1 \otimes \boldsymbol{y}_1, \boldsymbol{x}_1 \otimes \boldsymbol{y}_2, \ldots, \boldsymbol{x}_n \otimes \boldsymbol{y}_{q-1}, \boldsymbol{x}_n \otimes \boldsymbol{y}_q] \in \mathbb{R}^{mp \times nq}$.

$\boldsymbol{X} \odot \boldsymbol{Y}$ A Khatri-Rao product of two matrices $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{m \times n}$, $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_q] \in \mathbb{R}^{p \times q}$ with $n = q$, which is defined by $\boldsymbol{X} \odot \boldsymbol{Y} = [\boldsymbol{x}_1 \otimes \boldsymbol{y}_1, \boldsymbol{x}_2 \otimes \boldsymbol{y}_2, \ldots, \boldsymbol{x}_n \otimes \boldsymbol{y}_n]$. If $n = q = 1$, then $\boldsymbol{X} \otimes \boldsymbol{Y} = \boldsymbol{X} \odot \boldsymbol{Y}$ with size $mp \times n$.

$\mathcal{X} \times_d \boldsymbol{Y}$ A mode-d product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ with a matrix $\boldsymbol{Y} \in \mathbb{R}^{J \times I_d}$, $d = 1, 2, 3$, whose component is calculated by $(\mathcal{X} \times_1 \boldsymbol{Y})_{j i_2 i_3} = \sum_{i_1=1}^{I_1} x_{i_1 i_2 i_3} y_{j i_1}$, $(\mathcal{X} \times_2 \boldsymbol{Y})_{i_1 j i_3} = \sum_{i_2=1}^{I_2} x_{i_1 i_2 i_3} y_{j i_2}$, and $(\mathcal{X} \times_3 \boldsymbol{Y})_{i_1 i_2 j} = \sum_{i_3=1}^{I_3} x_{i_1 i_2 i_3} y_{j i_3}$.

$\boldsymbol{x} \circ \boldsymbol{y}$ A vector outer product of $\boldsymbol{x} \in \mathbb{R}^I$ and $\boldsymbol{y} \in \mathbb{R}^J$, whose component is calculated by $(\boldsymbol{x} \circ \boldsymbol{y})_{ij} = x_i y_j$.

$\text{prox}(\cdot)$ A proximal operator.

$\mathcal{S}(\cdot)$ A soft-thresholding operator.

1 Introduction

Many multistage manufacturing processes (MMPs) consist of multiple identical stages. For example, Figure 1 illustrates the layout of a hot rolling mill that is widely used in the steel-making industry. The primary function of the mill is to roll reheated semi-finished steel slabs thinner and longer through a series of rolling mill stands. The hot rolling mill in Figure 1 comprises seven identical stages (denoted as "Stage 1" to "Stage 7"). Another example of the MMP with identical stages is additive manufacturing (AM), which is a computer-controlled process that creates three-dimensional objects by depositing materials layer by layer (Gibson et al., 2021). In the AM process, each layer can be seen as one stage. For such a MMP, since all the stages are identical, they have the same process variables (i.e., process control parameters or sensing signals), which generate multi-channel time-series signals (referred to as "process data" hereafter). For instance, each of the stages in the hot rolling mill in Figure 1 generates multi-channel process data from the following process variables: *target speed of rollers*, *measured speed of rollers*, *looper value*, *target force on both sides of the rollers*, *measured force on the work side of rollers*, *measured force on the transfer side of rollers*, *roller gap*, *looper height*, and *temperature*, etc (Jeong and Fang, 2022).

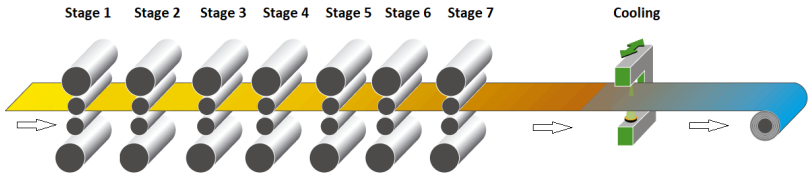
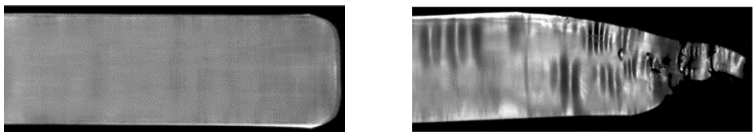


Figure 1: A steel slab rolling mill with seven stands (Jeong and Fang, 2022).

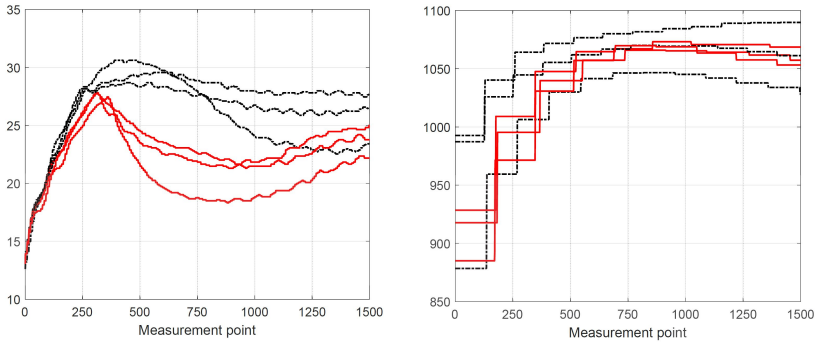
The multi-channel process data from a MMP typically vary from one product to another and are associated with the quality of products. This is because when fabricating products, process control parameters are adjusted (in real-time) by a closed-loop feedback control algorithm for product quality guarantees, which results in the change of process data. Also, the poor performance and/or lack of robustness of the control algorithm leads to inappropriate process control parameter values that result in product quality defects. Figure 2 illustrates products without and with quality defects obtained from the hot rolling mill shown in Figure 1. Figure 3 presents some example process data obtained from two out of

multiple process variables for six products (three defective and three non-defective). Since the multi-channel process data are associated with product quality defects, they can be used for the defect root cause diagnosis, which focuses on identifying the crucial process variables (as well as their stage locations) whose inappropriate values are responsible for the quality defect of products. Quality defect diagnosis plays an important role in product quality control since diagnostic results can be used to guide the modification of the feedback control algorithm to reduce the probability of fabricating defective products in the future.



(a) A non-defective product (b) A defective product

Figure 2: Products without and with quality defects from a hot rolling process (Balmashnova et al., 2013).



(a) Looper height (b) Temperature

Figure 3: Illustration of multi-channel process data obtained from the looper height variable at Stage 4 and the temperature variable at Stage 1 with six products (three black dash-dotted lines and three red solid lines represent non-defective and defective products, respectively).

Quality defect diagnosis can usually be achieved by employing data science methods that establish a mapping between a product's quality index and its multi-channel process data. This approach enables the identification of the subset of critical process variables (and their stage locations) that exhibit a strong association with product quality defects. Among the large number of fault diagnosis methods, penalization/regularization-based regression is a systematical approach with well-established statistical properties. For example, to identify

the crucial process variables (and their stage locations) that are responsible for product quality defects in the hot rolling mill in Figure 1, a straightforward method is to build a logistic regression model that maps a product's quality index (a binary variable, which is "1" if the product is defective and "0" otherwise) against its process data and penalize the sum of ℓ_2 norms of the regression coefficients corresponding to each process variable (or each stage) (Meier et al., 2008). Any process variables (and their stage locations) whose coefficients are penalized to be zeros are identified as non-crucial variables (and stages), and other process variables (and stages) with non-zero coefficients are selected to be responsible for the quality defect of the products. However, such a penalized logistic regression model has the following three limitations.

First, the number of unknown parameters in the model is extremely huge but the number of samples for model training is usually limited, so both estimated regression coefficients and diagnostic results are neither stable nor reliable. As an example, Figure 4 illustrates the structure of the process data for the hot rolling mill, which has seven stages, nine process variables in each stage, and $1,500$ measurement points over time for each process variable at each stage. Therefore, the number of elements of the process data corresponding to each product is $7 \times 9 \times 1,500 = 94,500$. This implies that the number of unknown parameters to be estimated in the logistic regression model is $94,501$ (1 for the intercept term). However, the maximum number of samples available for model training is usually not larger than several hundred, which results in the "large p, small n" problem in statistical estimation and inference.

Second, the time-series data from some of the process variables and stages are often highly correlated, which compromises the accuracy of diagnostic results. It is known that many penalization-based variable selection methods are sensitive to the correlation among predictors (Tibshirani, 1996; Yuan and Lin, 2006; Meier et al., 2008). Thus, it is beneficial to reduce or remove the high correlation among predictors (e.g., process variables, stages, and measurement points in the hot rolling mill application) when conducting variable selection.

Third, the penalized logistic regression proposed by Meier et al. (2008) cannot provide a structured variable selection solution. Specifically, since it separately penalizes the ℓ_2 norm of the regression coefficients corresponding to each process variable at each stage (each process variable has multiple coefficients since it has multiple measurement points),

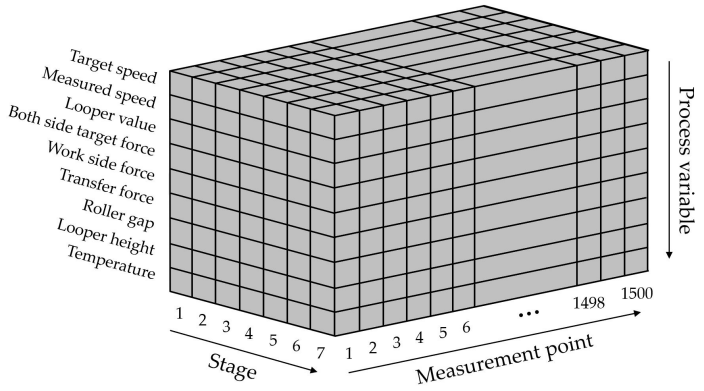
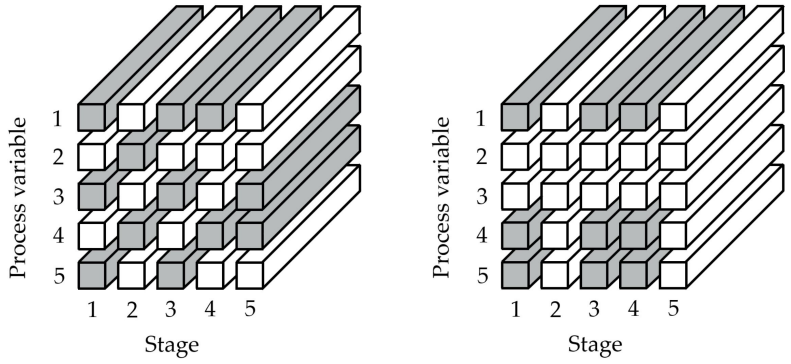


Figure 4: Data structure of the steel slab process data obtained from a hot rolling mill (the first dimension represents the process variable, the second dimension is the stage, and the third dimension is the measurement point).

it provides an unstructured diagnostic result that suggests various combinations of process variables and stages are responsible for product quality defects. Figure 5(a) shows an example of such an unstructured solution, in which the fibers with dark color are process variables selected as crucial. It can be seen that the first process variable is identified as crucial in Stage 1 but non-informative in Stage 2, while the second process variable is identified as important in Stage 2 but not Stage 1. Such an unstructured result misguides engineers in the revision of the feedback control algorithm (Jeong and Fang, 2022) since it selects different process variables at different stages. What control algorithm engineers prefer is a structured solution that after removing the non-informative variables and stages, all the remaining process variables and their stage locations are considered crucial for the product quality defects. Figure 5(b) illustrates such a structured solution, in which Process variables 1, 4, and 5 as well as Stages 1, 3, and 4 are considered important. To the best of our knowledge, there is no existing work that is able to address all three aforementioned limitations.

1.1 *Related Work*

In the literature, there are several methods that can possibly be used to jointly identify the critical process variables and stage locations accountable for product quality defects (i.e., addressing the third limitation by providing a structured diagnostic result) (Zhao and Leng, 2014; Zhao et al., 2017; Jeong and Fang, 2022). For example, the authors in



(a) An unstructured solution (b) A structured solution

Figure 5: An example of unstructured and structured solutions (the colored fibers are informative).

Zhao and Leng (2014) proposed the structured LASSO, which maps the expectation of a normally distributed response variable to an explanatory matrix using a bilinear product and penalizes the regression coefficients. Zhao et al. (2017) proposed a trace regression model that regresses a Gaussian response variable against its 2D explanatory matrix. To achieve 2D variable selection, it penalizes both rows and columns of the coefficient matrix simultaneously using the group LASSO penalty. In Jeong and Fang (2022), the authors developed a new 2D variable selection method based on a penalized matrix regression model, which regresses the quality index of a product against its process variable matrix. Here, the unknown regression coefficient matrix is decomposed as the product of two factor matrices, and the rows of the first factor matrix and columns of the second matrix are penalized simultaneously using the sum of ℓ_2 norms to inspire sparsity.

Although extensive numerical studies have shown that the aforementioned models work relatively well, they all require the process data to be a matrix. This implies that, to use these methods, we will have to transform the 3D process data such as in Figure 4 into a matrix form. This is typically done by taking the average of the observations at multiple measurement points of each process variable (at each stage location) and using its mean value to represent the whole sequence of observations (i.e., eliminating the measurement point dimension in Figure 4). One obvious limitation of doing so is the loss of useful information, and thus the accuracy of diagnostic results is compromised.

To preserve the information in process variables when conducting quality defect diagnosis, we may model the process data as a tensor. For example, the process data in Figure

4 is a 3D tensor. Diagnosis can be conducted by constructing a tensor regression model that regresses the quality index of a product against its process data of a tensor form and penalizes the regression coefficients. Many tensor regression models have been developed in the literature (Zhou et al., 2013; Hoff, 2015; Fang et al., 2019; Gahrooei et al., 2019; Yue et al., 2020; Wang et al., 2021; Miao et al., 2022; Gaw et al., 2022; Zhao et al., 2023; Zhou and Fang, 2023; Zhou et al., 2023). For example, Yue et al. (2020) proposed a tensor mixed effects model to effectively analyze and separate mixed effects in massive high-dimensional Raman mapping data used in nanomanufacturing quality inspections. Shen et al. (2022) introduced a new super-resolution method using smooth and sparse tensor completion to integrate image stream data from various imaging systems with complementary resolutions, enhancing both spatial and temporal resolution for more effective quality control. Shen et al. (2022) developed a smooth robust tensor completion model that effectively combines video recovery and background/foreground separation into a unified framework, addressing the challenge of missing pixels in robust tensor PCA. Although these models have demonstrated superior performance, they are not suitable for the applications discussed in this article for the following reasons. First, although some works use regularization terms to inspire sparsity, none of them can yield a structured solution for variable selection as the one shown in Figure 5(b). This implies that they cannot be used to jointly identify the important process variables and their stage locations. Second, although these works employ or develop particular optimization algorithms for parameter estimation, most of them are iterative algorithms without analytical or closed-form solutions. Thus, they will have to utilize certain optimization software for parameter estimation, which is typically very expensive in computation and thus not suitable for large-scale datasets.

1.2 *Contributions of this Article*

This study proposes a new tensor-based diagnosis method that simultaneously identifies the crucial process variables and their stage locations responsible for product quality defects.

The proposed methods regress the quality index of a product against its process tensor data, where the quality index follows an exponential family of distributions. To address the challenge of estimating a large number of unknown parameters with a relatively limited number of historical data samples (i.e., the "large p small n" problem), we decompose

the unknown tensor coefficients using the CANDECOMP/PARAFAC (CP) (Carroll and Chang, 1970) and Tucker (Tucker, 1966) decompositions, where CP expands the coefficient tensor as a product of several low-dimensional basis matrices, and Tucker decomposes the coefficient as a product of a core tensor and several low-dimensional factor matrices. By employing one of the decomposition methods, we estimate the basis/factor matrices (and a core tensor if Tucker is chosen) instead of estimating the high-dimensional coefficient tensor itself. Thus, it significantly reduces the number of parameters to be estimated and thus reduces the number of historical data samples needed for model estimation. Another benefit of using the tensor decompositions is that they can reduce the high correlation among process variables, stages, and measurement points such that diagnostic accuracy can be improved (i.e., addressing the aforementioned high correlation challenge). This is because applying the CP/Tucker decomposition to the unknown coefficient tensor is equivalent to applying dimensionality reduction to the process tensor data (also means removing some of the correlation in process variables, stages, and measurement points). In the literature, the tensor train (TT) decomposition (Oseledets, 2011) is another frequently employed method for tensor decomposition in regression analysis. It represents a tensor as a sequence of interconnected lower-dimensional (core) tensors arranged in a train-like structure. Each core tensor in this sequence is linked to its neighbors through matrix multiplications along one mode, facilitating an efficient and scalable representation of high-dimensional data, thereby mitigating the curse of dimensionality. In this article, we choose CP and Tucker decompositions over TT decomposition because designing a regularization term for TT decomposition that yields structured diagnostic results is infeasible.

To achieve the goal of jointly identifying the crucial process variables and stages, we incorporate the specially designed regularization term into the tensor regression model. To be specific, we simultaneously penalize the rows of the first and second basis/factor matrices using the sum of ℓ_2 norms. In addition, to estimate the parameters, we first propose the block coordinate descent algorithm that cyclically updates each block of parameters. We then design the block coordinate proximal descent algorithm, which exploits closed-form solutions. We also prove that both of the two optimization algorithms possess the global convergence property, which implies that they converge to a critical point of the optimization criterion from any initial point.

The proposed methods are applicable to a broad range of industrial applications that utilize multiple identical manufacturing stages, extending beyond the previously discussed hot rolling for strip steel and additive manufacturing. Examples include paper manufacturing, where identical drying cylinders sequentially remove moisture to consistently form and dry paper. In glass manufacturing, identical annealing lehrs uniformly cool glass sheets to prevent defects. Aluminum rolling features several identical rolling mills in the finishing stage to precisely reduce thickness, akin to steel rolling. Moreover, high-volume printing operations employ identical printing units to apply various ink layers sequentially, ensuring consistent, high-quality prints. These examples highlight the critical role of multiple identical machines in achieving product uniformity and adhering to quality standards across different industries, which stand to benefit from the proposed diagnosis methods.

The rest of the article is organized as follows. Section 2 discusses the tensor-based quality fault diagnosis methodology. Section 3 presents optimization algorithms for parameter estimation. Sections 4 and 5 evaluate the performance of our proposed methods using simulated and real-world datasets, respectively. Section 6 provides concluding remarks.

2 The Tensor-based Quality Fault Diagnosis Methodology

2.1 *Generalized Linear Models (GLMs)*

Suppose there exists a dataset that consists of quality indices and process data of n products obtained from a MMP. Let $y_i \in \mathbb{R}$ denote the quality index and $\mathcal{X}_i \in \mathbb{R}^{P_1 \times P_2 \times P_3}$ the process variable tensor of product i for $i = 1, \ldots, n$, where P_1 is the number of process variables, P_2 is the number of stages, and P_3 is the number of measurement points. We assume that Y_i is a random variable with an independent observation y_i from a distribution in the exponential family. Thus, its probability mass or density function can be expressed as follows (McCullagh and Nelder, 1989):

$$f(y_i|\theta_i, \phi) = \exp\left[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right], \tag{1}$$

where θ_i is the natural parameter and $\phi > 0$ is the dispersion parameter. $a(\cdot)$, $b(\cdot)$, and $c(\cdot)$

are known functions determined by the specific distribution in the exponential family. For example, if Y_i follows a normal distribution with mean μ_i and standard deviation σ, i.e., $Y_i \sim N(\mu_i, \sigma^2)$, then the parameters and functions are described as $\theta_i = \mu_i$, $\phi = \sigma^2$, $a(\phi) = \phi$, $b(\theta_i) = \theta_i^2/2$, and $c(y_i, \phi) = -\frac{1}{2}(y_i^2/\phi + \log(2\pi\phi))$; if Y_i follows a binomial distribution with n_i samples and the probability of a target quality index occurring, π_i, i.e., $y_i \sim B(n_i, \pi_i)$, then the parameters and functions can be represented as $\theta_i = \log(\pi_i/(1-\pi_i))$, $a(\phi) = 1$, $b(\theta_i) = n_i \log(1 + e^{\theta_i})$, and $c(y_i, \phi) = \log \binom{n_i}{y_i}$. In this article, we mainly consider the GLMs in which the dispersion parameter is known. Examples include Bernoulli, binomial, Poisson, and exponential distributions.

The relationship between quality indices and process tensor data can be established by using a known link function $g(\cdot)$, which links the expected value of Y_i, i.e., $\mu_i = \mathbb{E}[Y_i]$ to the linear combination of predictors, $\alpha + \langle \mathcal{B}, \mathcal{X}_i \rangle$ as follows:

$$g(\mu_i) = \alpha + \langle \mathcal{B}, \mathcal{X}_i \rangle, \tag{2}$$

where $\alpha \in \mathbb{R}$ is the intercept, $\mathcal{B} \in \mathbb{R}^{P_1 \times P_2 \times P_3}$ is the unknown regression coefficient tensor, and $\langle \cdot, \cdot \rangle$ is the element-wise inner product operator of two tensors. As an example of the link function $g(\cdot)$, when Y_i follows a binomial distribution, one of the choices for $g(\cdot)$ is the logit function, namely, $g(\mu_i) = \log(\mu_i/(1-\mu_i))$.

The coefficient tensor $\mathcal{B}$ in Equation (2) can be estimated by minimizing the loss function $\mathcal{L}$, i.e., $\min_{\mathcal{B},\alpha} \ \mathcal{L}(\mathcal{B}, \alpha)$, where the loss function is the negative log-likelihood function if the response variable (i.e., quality index) follows a Bernoulli, binomial, Poisson, or exponential distribution. If it follows a normal distribution, we use the squared error loss for the loss function.

As discussed in Section 1, one of the challenges is that the number of parameters to be estimated is large, whereas the number of data samples is relatively small. To address this challenge, we expand the high-dimensional unknown coefficient tensor $\mathcal{B}$ using the CP/Tucker decomposition, which provides a set of low-dimensional basis/factor matrices and a core tensor whose size is much smaller than that of $\mathcal{B}$. Instead of estimating the high-dimensional coefficient tensor itself, we estimate the basis/factor matrices and the core tensor, which significantly reduces the number of parameters to be estimated.

2.2 Coefficient Expansion using Tensor Decompositions

The *CP decomposition* expands the unknown coefficient tensor $\mathcal{B}$ as a product of several basis matrices:

$$\mathcal{B} \approx \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \boldsymbol{\beta}_2^{(r)} \circ \boldsymbol{\beta}_3^{(r)} \equiv [\![\boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3]\!], \tag{3}$$

where R is the rank of the CP decomposition determined by certain model selection criteria such as AICc (to be discussed later), and $\boldsymbol{\beta}_d^{(r)} = [\beta_{d,1}^{(r)}, \ldots, \beta_{d,P_d}^{(r)}]^\top \in \mathbb{R}^{P_d}$. The operator "$\circ$" denotes the outer product. Also, it can be shown that $\text{vec}(\mathcal{B}) \approx (\boldsymbol{B}_3 \odot \boldsymbol{B}_2 \odot \boldsymbol{B}_1)\mathbf{1}_R$, where $\boldsymbol{B}_d = [\boldsymbol{\beta}_d^{(1)}, \ldots, \boldsymbol{\beta}_d^{(R)}] \in \mathbb{R}^{P_d \times R}$ for $d = 1, 2, 3$, denotes the basis matrix, $\mathbf{1}_R \in \mathbb{R}^R$ is a vector of ones with size R, and the operator "$\odot$" represents the *Khatri-Rao* product (Kolda and Bader, 2009; Fang et al., 2019).

The *Tucker decomposition* expands the coefficient tensor $\mathcal{B}$ into one core tensor and a set of factor matrices:

$$\mathcal{B} \approx \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1,r_2,r_3} \boldsymbol{\beta}_1^{(r_1)} \circ \boldsymbol{\beta}_2^{(r_2)} \circ \boldsymbol{\beta}_3^{(r_3)} \equiv [\![\mathcal{G}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3]\!], \tag{4}$$

where $g_{r_1,r_2,r_3} = (\mathcal{G})_{r_1,r_2,r_3}$ is the (r_1, r_2, r_3)-entry of the core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$, and $\boldsymbol{B}_d = [\boldsymbol{\beta}_d^{(1)}, \ldots, \boldsymbol{\beta}_d^{(R_d)}] \in \mathbb{R}^{P_d \times R_d}$ for $d = 1, 2, 3$, is the factor matrix. It is known that $\text{vec}(\mathcal{B}) \approx \mathcal{G} \times_1 \boldsymbol{B}_1 \times_2 \boldsymbol{B}_2 \times_3 \boldsymbol{B}_3$ holds, where the operator "$\times_d$" stands for the mode-d product (Kolda and Bader, 2009; Fang et al., 2019).

As a result, the regression model in Equation (2) can be expressed as follows:

$$\begin{aligned} g(\mu_i) &= \alpha + \left\langle \mathcal{B}, \mathcal{X}_i \right\rangle \\ &= \alpha + \left\langle \text{vec}(\mathcal{B}), \text{vec}(\mathcal{X}_i) \right\rangle \\ &\approx \begin{cases} \alpha + \left\langle (\boldsymbol{B}_3 \odot \boldsymbol{B}_2 \odot \boldsymbol{B}_1)\mathbf{1}_R, \text{vec}(\mathcal{X}_i) \right\rangle, & if \text{ CP}, \\ \alpha + \left\langle \mathcal{G} \times_1 \boldsymbol{B}_1 \times_2 \boldsymbol{B}_2 \times_3 \boldsymbol{B}_3, \text{vec}(\mathcal{X}_i) \right\rangle, & if \text{ Tucker}. \end{cases} \end{aligned} \tag{5}$$

Equation (5) implies that instead of estimating $\mathcal{B} \in \mathbb{R}^{P_1 \times P_2 \times P_3}$ with $P_1 \times P_2 \times P_3$ parameters, we can estimate $\{\boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3\}$ with $(P_1 + P_2 + P_3) \times R$ parameters if CP is employed or $\{\mathcal{G}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3\}$ with $R_1R_2R_3 + P_1R_1 + P_2R_2 + P_3R_3$ parameters if Tucker is used. Since the rank is usually low, this helps significantly reduce the number of parameters to be esti-

mated. Taking the hot rolling mill in Figure 1 as an example, $P_1 = 9, P_2 = 7, P_3 = 1,500$, which implies that the coefficient tensor $\mathcal{B}$ has $94,500 = 9 \times 7 \times 1,500$ elements to be estimated. If the rank $R = 2$ for CP and $(R_1, R_2, R_3) = (2, 1, 2)$ for Tucker, the number of elements in the coefficient tensor to be estimated is reduced to $3,032 = (9 + 7 + 1,500) \times 2$ and $3,030 = 2 \times 1 \times 2 + 9 \times 2 + 7 \times 1 + 1,500 \times 2$, respectively.

In real-world applications, the rank of the coefficient tensor $\mathcal{B}$ is usually low since there exists heavy correlation within the process tensor data (e.g., the observations from each process variable are auto-correlated and those from different process variables are cross-correlated). Expanding the coefficient tensor into low-dimensional basis/factor matrices and the core tensor using the CP/Tucker decomposition is equivalent to conducting dimensionality reduction on the process tensor data, which helps to remove or decrease the correlation among process variables, stages, and measurement points (Fang et al., 2019) and thus improve the accuracy and stability of subsequent diagnosis.

2.3 *Regularization and Model Selection*

To simultaneously identify the crucial process variables and their stage locations responsible for product quality defects, we add a regularization term to the loss function when conducting parameter estimation. Here, we demonstrate the Tucker-based method as an example to discuss the construction of the regularization term. The regularization term for the CP-based method is same as that for the Tucker-based method except that the core tensor $\mathcal{G}$ is excluded.

The regularization term is designed to penalize the rows of the first and second basis/factor matrices $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ using the sum of ℓ_2 norms. In addition, we penalize the core tensor $\mathcal{G}$ and the third basis/factor matrix $\boldsymbol{B}_3$ using ℓ_1 norms to enhance the numerical stability in parameter estimation and alleviate possible non-uniqueness (non-identifiability) of parameter estimates. More details about the identifiability issue in the tensor regression setting are discussed in Zhou et al. (2013). As a result, parameter estimation is performed by solving the following optimization problem:

$$\min_{\mathcal{G}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3, \alpha} \mathcal{F}(\mathcal{G}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3, \alpha) \coloneqq \mathcal{L}(\mathcal{G}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3, \alpha) + \mathcal{R}(\mathcal{G}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3), \tag{6}$$

where the regularization term is defined as follows:

$$\mathcal{R}(\mathcal{G}, \boldsymbol{B}_1, \boldsymbol{B}_2, \boldsymbol{B}_3) = \lambda \left[\|\mathcal{G}\|_1 + \sum_{p_1=1}^{P_1} \gamma_1 \|\boldsymbol{b}_{1,p_1}\|_2 + \sum_{p_2=1}^{P_2} \gamma_2 \|\boldsymbol{b}_{2,p_2}\|_2 + \|\boldsymbol{B}_3\|_1 \right]. \tag{7}$$

Here, $\lambda \geq 0$ is a tuning parameter and $\|\cdot\|_q$ is the ℓ_q norm. $\boldsymbol{b}_{1,p_1} \in \mathbb{R}^{1\times R_1}$ is the $p_1 th$ row of $\boldsymbol{B}_1 \in \mathbb{R}^{P_1 \times R_1}$ and $\boldsymbol{b}_{2,p_2} \in \mathbb{R}^{1\times R_2}$ is the $p_2 th$ row of $\boldsymbol{B}_2 \in \mathbb{R}^{P_2 \times R_2}$. In the CP-based method, $R_1 = R_2 = R$ is the rank of $\mathcal{B}$, and in the Tucker-based method, R_1 and R_2 are respectively the first and second components of the rank $\{R_d\}_{d=1}^3$ of $\mathcal{B}$. $\gamma_1 = \sqrt{R_1}$ and $\gamma_2 = \sqrt{R_2}$ are used to rescale the penalty terms $\|\boldsymbol{b}_{1,p_1}\|_2$ and $\|\boldsymbol{b}_{2,p_2}\|_2$, respectively, since the lengths of vectors $\boldsymbol{b}_{1,p_1}$ and $\boldsymbol{b}_{2,p_2}$ are different. These values ensure that the vectors with different lengths are penalized more fairly (Yuan and Lin, 2006). Such a scaling is particularly important for the Tucker-based method because the number of elements in the two penalty terms can be significantly different when the two components of the rank, R_1 and R_2, are different.

The tuning parameter λ and rank $\{R_d\}_{d=1}^3$ (or R for the CP-based method) can be selected using a model selection criterion such as AIC and BIC. Both Akaike Information Criterion (AIC) (Akaike, 1974) and Bayesian Information Criterion (BIC) (Schwarz, 1978) evaluate the goodness of fit for a candidate model by striking a balance between its likelihood and model complexity, which is determined by the number of parameters. The difference between these two criteria lies in their respective degrees of penalizing model complexity. Specifically, AIC puts less penalty to model complexity, leading to the selection of more complex model than BIC. That is, AIC may be favored when pursuing a more flexible model, whereas BIC typically favors a more parsimonious model. Both AIC and BIC have been extensively utilized in the literature without a particular preference (Kutner et al., 2005; Faraway, 2014).

In this article, we use the second-order Akaike Information Criterion (AICc) (Hurvich and Tsai, 1989) as a model selection criterion. AICc is the bias-corrected AIC that adds a small-sample-size bias correction term to AIC. It offers advantages over AIC and BIC in our diagnostic application, particularly for small sample sizes. It includes a correction term that adjusts the penalty for model complexity, reducing the bias that AIC might introduce and avoiding the tendency of BIC to favor overly complex models in small datasets. AICc

also supports reliable model selection for a range of sample sizes, making it well-suited for the variable datasets typical in real-world multistage manufacturing processes, especially given the large number of parameters that need to be estimated. AICc is defined as follows:

$$\begin{aligned} AICc &= AIC + \frac{2p(p+1)}{n-p-1} \\ &= -2\ell(\hat{\mathcal{G}}, \hat{\boldsymbol{B}}_1, \hat{\boldsymbol{B}}_2, \hat{\boldsymbol{B}}_3, \hat{\alpha}) + 2p + \frac{2p(p+1)}{n-p-1}, \end{aligned} \tag{8}$$

where $\ell(\hat{\mathcal{G}}, \hat{\boldsymbol{B}}_1, \hat{\boldsymbol{B}}_2, \hat{\boldsymbol{B}}_3, \hat{\alpha})$ is the log-likelihood function value that is evaluated at the solution $(\hat{\mathcal{G}}, \hat{\boldsymbol{B}}_1, \hat{\boldsymbol{B}}_2, \hat{\boldsymbol{B}}_3, \hat{\alpha})$ of the optimization problem (6). n is the sample size, and p is the number of nonzero parameters. When $n \to \infty$, the bias-corrected term converges to 0 and thus AICc converges to AIC. This implies that if the ratio n/p is sufficiently large, then AIC and AICc will be similar and tend to select the same model. In general, the use of AICc is recommended when the ratio n/p is small, say < 40 (Burnham and Anderson, 2002).

The accurate rank selection is important to the performance of the proposed methods and many other tensor decomposition-based statistical learning methods (Fang et al., 2019). However, the given sample size is often limited in reality, so it is expected that a lower rank will be selected when the true rank is high. This is because a higher rank implies more parameters to be estimated, which requires more samples for model training; also, similar to many other model selection criterion, AIC and AICc are data-driven, so they will select a rank that yields the best performance of the model, which might not be a rank close to the true one. In Section 4, we will show that the proposed AICc criterion works relatively well in identifying an appropriate rank for the proposed methods.

We set the same tuning parameter λ for all terms in (7) for ease of implementation. At the cost of computation efficiency, it is possible to set individual tuning parameters for each term.

3 Optimization Algorithms for Parameter Estimation

In this section, we discuss the optimization algorithms to solve optimization criterion in (6). We will again use the Tucker-based model as an example. For the CP-based method, the optimization algorithms are same as those of the Tucker-based method except that the core

tensor $\mathcal{G}$ needs to be excluded.

To solve the optimization problem (6), we first propose the general block coordinate descent (BCD) algorithm (summarized in Algorithm 1), which cyclically optimizes one block of the parameters while keeping other blocks constant.

Algorithm 1 Block Coordinate Descent (BCD)

Input: Data $\{\mathcal{X}_i, y_i\}_{i=1}^n$ and rank $\begin{cases} R & if \text{ CP}, \\ \{R_d\}_{d=1}^3 & if \text{ Tucker}. \end{cases}$

Initialization: Randomly choose $\{\mathcal{B}^0, \alpha^0\}$. $k \leftarrow 1$.

Decompose $\mathcal{B}^0$ using one of the tensor decompositions:

$$\begin{cases} \{\boldsymbol{B}_1^0, \boldsymbol{B}_2^0, \boldsymbol{B}_3^0\} \leftarrow \texttt{CPDecomp}(\mathcal{B}^0) & if \text{ CP}, \\ \{\mathcal{G}^0, \boldsymbol{B}_1^0, \boldsymbol{B}_2^0, \boldsymbol{B}_3^0\} \leftarrow \texttt{TuckerDecomp}(\mathcal{B}^0) & if \text{ Tucker}. \end{cases}$$

while convergence criterion not met **do**

  $\mathcal{G}^k \leftarrow \operatorname{argmin}_{\mathcal{G}} \mathcal{F}(\mathcal{G}, \boldsymbol{B}_1^{k-1}, \boldsymbol{B}_2^{k-1}, \boldsymbol{B}_3^{k-1}, \alpha^{k-1})$.

  $\boldsymbol{B}_1^k \leftarrow \operatorname{argmin}_{\boldsymbol{B}_1} \mathcal{F}(\mathcal{G}^k, \boldsymbol{B}_1, \boldsymbol{B}_2^{k-1}, \boldsymbol{B}_3^{k-1}, \alpha^{k-1})$.

  $\boldsymbol{B}_2^k \leftarrow \operatorname{argmin}_{\boldsymbol{B}_2} \mathcal{F}(\mathcal{G}^k, \boldsymbol{B}_1^k, \boldsymbol{B}_2, \boldsymbol{B}_3^{k-1}, \alpha^{k-1})$.

  $\boldsymbol{B}_3^k \leftarrow \operatorname{argmin}_{\boldsymbol{B}_3} \mathcal{F}(\mathcal{G}^k, \boldsymbol{B}_1^k, \boldsymbol{B}_2^k, \boldsymbol{B}_3, \alpha^{k-1})$.

  $\alpha^k \leftarrow \operatorname{argmin}_{\alpha} \mathcal{F}(\mathcal{G}^k, \boldsymbol{B}_1^k, \boldsymbol{B}_2^k, \boldsymbol{B}_3^k, \alpha)$.

  $k \leftarrow k + 1$.

end while

$\hat{\mathcal{G}} \leftarrow \mathcal{G}^k$, $\hat{\boldsymbol{B}}_1 \leftarrow \boldsymbol{B}_1^k$, $\hat{\boldsymbol{B}}_2 \leftarrow \boldsymbol{B}_2^k$, $\hat{\boldsymbol{B}}_3 \leftarrow \boldsymbol{B}_3^k$, $\hat{\alpha} \leftarrow \alpha^k$.

Output: $\hat{\mathcal{G}}, \hat{\boldsymbol{B}}_1, \hat{\boldsymbol{B}}_2, \hat{\boldsymbol{B}}_3, \hat{\alpha}$.

Next, we extend the BCD algorithm to the block coordinate proximal descent (BCPD) algorithm, which exploits closed-form solutions by considering the non-differentiability of the regularization terms in (7). Specifically, Theorem 1 suggests that if the negative log-likelihood function has a Lipschitz continuous gradient, the (proximal) gradient descent step for each sub-problem in Algorithm 1 has closed-form solutions.

Theorem 1. *If the response variable Y_i follows a distribution whose negative log-likelihood function $\mathcal{L}$ has a Lipschitz continuous gradient, then BCPD has the following closed-form*

solutions to solve the optimization problem (6).

$$
\begin{aligned}
\mathcal{G}^k &= \mathcal{S}^{(1)}_{\lambda\tau_0^k}\big(\mathcal{G}^{k-1} - \tau_0^k \nabla_{\mathcal{G}} \mathcal{L}(\mathcal{G}^{k-1}, \boldsymbol{B}_1^{k-1}, \boldsymbol{B}_2^{k-1}, \boldsymbol{B}_3^{k-1}, \alpha^{k-1})\big),\\
\boldsymbol{B}_1^k &= \mathcal{S}^{(2)}_{\lambda\gamma_1\tau_1^k}\big(\boldsymbol{B}_1^{k-1} - \tau_1^k \nabla_{\boldsymbol{B}_1} \mathcal{L}(\mathcal{G}^{k}, \boldsymbol{B}_1^{k-1}, \boldsymbol{B}_2^{k-1}, \boldsymbol{B}_3^{k-1}, \alpha^{k-1})\big),\\
\boldsymbol{B}_2^k &= \mathcal{S}^{(2)}_{\lambda\gamma_2\tau_2^k}\big(\boldsymbol{B}_2^{k-1} - \tau_2^k \nabla_{\boldsymbol{B}_2} \mathcal{L}(\mathcal{G}^{k}, \boldsymbol{B}_1^{k}, \boldsymbol{B}_2^{k-1}, \boldsymbol{B}_3^{k-1}, \alpha^{k-1})\big),\\
\boldsymbol{B}_3^k &= \mathcal{S}^{(1)}_{\lambda\tau_3^k}\big(\boldsymbol{B}_3^{k-1} - \tau_3^k \nabla_{\boldsymbol{B}_3} \mathcal{L}(\mathcal{G}^{k}, \boldsymbol{B}_1^{k}, \boldsymbol{B}_2^{k}, \boldsymbol{B}_3^{k-1}, \alpha^{k-1})\big),\\
\alpha^k &= \alpha^{k-1} - \tau_4^k \nabla_{\alpha} \mathcal{L}(\mathcal{G}^{k}, \boldsymbol{B}_1^{k}, \boldsymbol{B}_2^{k}, \boldsymbol{B}_3^{k}, \alpha^{k-1}),
\end{aligned}
\tag{9}
$$

where $\gamma_1 = \sqrt{R_1}, \gamma_2 = \sqrt{R_2}$. $\tau_j^k > 0$ *for* $j = 0, \ldots, 4$ *is a step size.* $\mathcal{S}^{(1)}_{\lambda\tau_j^k}(\cdot)$ *for* $j = 0, 3$ *is the component-wise soft-thresholding operator, and* $\mathcal{S}^{(2)}_{\lambda\gamma_j\tau_j^k}(\cdot)$ *for* $j = 1, 2$ *is the soft-thresholding operator, which are defined by*

$$
\big[\mathcal{S}^{(1)}_{\lambda\tau_0^k}(\mathcal{G})\big]_{r_1,r_2,r_3} = \begin{cases} g_{r_1,r_2,r_3} - \lambda\tau_0^k, & if\ g_{r_1,r_2,r_3} > \lambda\tau_0^k \\ g_{r_1,r_2,r_3} + \lambda\tau_0^k, & if\ g_{r_1,r_2,r_3} < -\lambda\tau_0^k \\ 0, & if\ |g_{r_1,r_2,r_3}| \le \lambda\tau_0^k \end{cases} \tag{10}
$$

$$
\big[\mathcal{S}^{(2)}_{\lambda\gamma_j\tau_j^k}(\boldsymbol{B}_j)\big]_{p_j} = \begin{cases} \boldsymbol{b}_{j,p_j} - \lambda\gamma_j\tau_j^k \dfrac{\boldsymbol{b}_{j,p_j}}{\|\boldsymbol{b}_{j,p_j}\|_2}, & if\ \|\boldsymbol{b}_{j,p_j}\|_2 > \lambda\gamma_j\tau_j^k \\ \boldsymbol{0}, & if\ \|\boldsymbol{b}_{j,p_j}\|_2 \le \lambda\gamma_j\tau_j^k \end{cases} \tag{11}
$$

$$
\big[\mathcal{S}^{(1)}_{\lambda\tau_3^k}(\boldsymbol{B}_3)\big]_{p_3,r_3} = \begin{cases} b_{p_3,r_3} - \lambda\tau_3^k, & if\ b_{p_3,r_3} > \lambda\tau_3^k \\ b_{p_3,r_3} + \lambda\tau_3^k, & if\ b_{p_3,r_3} < -\lambda\tau_3^k \\ 0, & if\ |b_{p_3,r_3}| \le \lambda\tau_3^k \end{cases} \tag{12}
$$

where $\big[\mathcal{S}^{(1)}_{\lambda\tau_0^k}(\mathcal{G})\big]_{r_1,r_2,r_3}$ *denotes the* (r_1, r_2, r_3)*-entry of* $\mathcal{S}^{(1)}_{\lambda\tau_0^k}(\mathcal{G})$ *for* $r_d = 1, \ldots, R_d$, $d = 1, 2, 3$. $\big[\mathcal{S}^{(2)}_{\lambda\gamma_j\tau_j^k}(\boldsymbol{B}_j)\big]_{p_j}$ *denotes the* p_j*th row of* $\mathcal{S}^{(2)}_{\lambda\gamma_j\tau_j^k}(\boldsymbol{B}_j)$ *for* $p_j = 1, \ldots, P_j$, $j = 1, 2$. $\big[\mathcal{S}^{(1)}_{\lambda\tau_3^k}(\boldsymbol{B}_3)\big]_{p_3,r_3}$ *denotes the* (p_3, r_3)*-entry of* $\mathcal{S}^{(1)}_{\lambda\tau_3^k}(\boldsymbol{B}_3)$ *for* $p_3 = 1, \ldots, P_3$ *and* $r_3 = 1, \ldots, R_3$.

The step size $\tau_j^k > 0,\ j = 0, \ldots, 4$ can be set to a sufficiently small constant, a diminishing sequence (e.g., $1/k$), or determined by the backtracking line search (Boyd and Vandenberghe, 2004; Nocedal and Wright, 2006). In the simulation study, we set $\tau_j^k = 10^{-3}, \forall j$ for ease of implementation. Algorithm 2 summarizes the BCPD algorithm.

Algorithm 2 Block Coordinate Proximal Descent (BCPD)

Input: Data $\{\mathcal{X}_i, y_i\}_{i=1}^n$ and rank $\begin{cases} R & if \text{ CP}, \\ \{R_d\}_{d=1}^3 & if \text{ Tucker}. \end{cases}$

Initialization: Randomly choose $\{\mathcal{B}^0, \alpha^0\}$. $k \leftarrow 1$.

Decompose $\mathcal{B}^0$ using one of the tensor decompositions:

$$\begin{cases} \{\boldsymbol{B}_1^0, \boldsymbol{B}_2^0, \boldsymbol{B}_3^0\} \leftarrow \texttt{CPDecomp}(\mathcal{B}^0) & if \text{ CP}, \\ \{\mathcal{G}^0, \boldsymbol{B}_1^0, \boldsymbol{B}_2^0, \boldsymbol{B}_3^0\} \leftarrow \texttt{TuckerDecomp}(\mathcal{B}^0) & if \text{ Tucker}. \end{cases}$$

while convergence criterion not met **do**

  Compute $(\mathcal{G}^k, \boldsymbol{B}_1^k, \boldsymbol{B}_2^k, \boldsymbol{B}_3^k, \alpha^k)$ using (9).

  $k \leftarrow k + 1$.

end while

  $\hat{\mathcal{G}} \leftarrow \mathcal{G}^k$, $\hat{\boldsymbol{B}}_1 \leftarrow \boldsymbol{B}_1^k$, $\hat{\boldsymbol{B}}_2 \leftarrow \boldsymbol{B}_2^k$, $\hat{\boldsymbol{B}}_3 \leftarrow \boldsymbol{B}_3^k$, $\hat{\alpha} \leftarrow \alpha^k$.

Output: $\hat{\mathcal{G}}, \hat{\boldsymbol{B}}_1, \hat{\boldsymbol{B}}_2, \hat{\boldsymbol{B}}_3, \hat{\alpha}$.

The termination condition for the BCD/BCPD algorithm can be that the difference between the consecutive objective function values is less than a small number ϵ, that is, $|\mathcal{F}^k - \mathcal{F}^{k-1}| < \epsilon$, where $\mathcal{F}^k = \mathcal{L}(\mathcal{G}^k, \boldsymbol{B}_1^k, \boldsymbol{B}_2^k, \boldsymbol{B}_3^k, \alpha^k) + \mathcal{R}(\mathcal{G}^k, \boldsymbol{B}_1^k, \boldsymbol{B}_2^k, \boldsymbol{B}_3^k)$. The tolerance ϵ can be set to a sufficiently small value. Another possible termination condition is that the iteration number of the algorithms attains the maximum number of iterations δ, which can be set to a sufficiently large value. In this study, we set $\epsilon = 10^{-3}$ and $\delta = 500$. Whichever condition is met first, the algorithms will be terminated. Note that the optimization criterion in (6) is nonconvex. Thus, we usually try multiple starting points and choose the estimation result that shows the lowest objective function value.

Theorem 2 indicates that the BCPD algorithm has a global convergence property, which implies that it converges to a critical point in the optimization problem (6) with any initialization.

Theorem 2 (Global convergence)**.** *The sequence generated by the proposed BCPD algorithm converges to a critical point in the optimization problem* (6)*.*

The proof of Theorems 1 and 2 can be found in the supplementary material. To analyze the scalability of the proposed algorithm, we explore its computational complexity. Consider a D-dimensional coefficient tensor $\mathcal{B}^0 \in \mathbb{R}^{P_1 \times \cdots \times P_D}$ processed by the proposed algorithms. The complexity of CP decomposition of $\mathcal{B}^0$ mainly hinges on the rank R, the number of dimensions D, and the size of each dimension P_d, resulting in a complexity of

$\mathcal{O}(DR\prod_{d=1}^{D} P_d)$ per iteration when using the alternating least squares (ALS) method, commonly employed for such decompositions (Battaglino et al., 2018). When all mode ranks $\{R_d\}_{d=1}^{D}$ are set equal to R, Tucker decomposition achieves a complexity comparable to that of CP decomposition by utilizing the higher-order orthogonal iteration (HOOI) algorithm, which also employs the alternating least squares (ALS) method (Kolda and Bader, 2009). Specifically, for a three-dimensional tensor ($D = 3$) considered in this article, the complexity of CP decomposition is $\mathcal{O}(R \times P_1 \times P_2 \times P_3)$ per iteration, and a comparable calculation applies to Tucker decomposition.

The computational complexity is also affected by the number of iterations required for the BCD/BCPD algorithm to converge, as determined by the convergence criteria outlined in Algorithms 1 and 2. Assuming the convergence criterion specifies that the optimality gap must be reduced to below a tolerance $\tilde{\epsilon}$, and considering there are no inner iterations within each block of the BCD/BCPD algorithm's outer while loop–where each block directly proceeds to a single gradient step–the number of iterations needed for convergence is approximately $\mathcal{O}(1/\tilde{\epsilon})$ due to the convex nature of the block's objective function. More detailed discussion of the convergence speed of BCD and BCPD can be found in Hong et al. (2017) and Jeong et al. (2023).

4 Simulation Study

In this section, we demonstrate the superiority of the proposed methods using synthetic datasets over other alternatives.

4.1 *Data Generation*

We first generate three-dimensional process data for n products with P_1 process variables, P_2 stages, and P_3 measurement points, denoted by $\{\mathcal{X}_i \in \mathbb{R}^{P_1 \times P_2 \times P_3}\}_{i=1}^{n}$. Here, we conduct experiments for the cases of $n = 200, 300, 500$ with $(P_1, P_2, P_3) = (7, 8, 5)$ and $(9, 10, 10)$. We also consider two types of correlation structures: (a) independent and identically distributed (i.i.d.) and (b) stage correlated cases. For the first structure, all the elements of $\mathcal{X}_i$ are generated from an i.i.d. standard normal distribution. For the second structure, we set the correlation between $\boldsymbol{x}_{:p_2p_3}$ and $\boldsymbol{x}_{:p_2'p_3}$ as $0.5^{|p_2-p_2'|}$ for every $p_3 \in \{1, \ldots, P_3\}$, where

$p_2, p_2' = 1, \ldots, P_2$ and $p_2 \neq p_2'$.

Next, we generate underlying regression coefficient tensors $\mathcal{B} \in \mathbb{R}^{P_1 \times P_2 \times P_3}$ for both the CP-based method (use a subscript "C") and the Tucker-based method (use a subscript "T"). The coefficient tensor $\mathcal{B}_C$ is constructed from three basis matrices $\boldsymbol{B}_{C,1} \in \mathbb{R}^{P_1 \times 3}$, $\boldsymbol{B}_{C,2} \in \mathbb{R}^{P_2 \times 3}$, and $\boldsymbol{B}_{C,3} \in \mathbb{R}^{P_3 \times 3}$ where $(P_1, P_2, P_3) = (7, 8, 5)$ and $(9, 10, 10)$, all the elements of which are randomly generated from a uniform distribution Unif$(-1, 1)$. We then let the odd rows of $\boldsymbol{B}_{C,1}$ and the even rows of $\boldsymbol{B}_{C,2}$ be zeros. In a similar manner, the coefficient tensor $\mathcal{B}_T$ is constructed from one core tensor and three factor matrices: $\mathcal{G}_T \in \mathbb{R}^{2 \times 1 \times 2}$, $\boldsymbol{B}_{T,1} \in \mathbb{R}^{P_1 \times 2}$, $\boldsymbol{B}_{T,2} \in \mathbb{R}^{P_2 \times 1}$, and $\boldsymbol{B}_{T,3} \in \mathbb{R}^{P_3 \times 2}$ where $(P_1, P_2, P_3) = (7, 8, 5)$ and $(9, 10, 10)$, all the elements of which are randomly generated from a uniform distribution Unif$(-1, 1)$. Similar to $\mathcal{B}_C$, the odd rows of $\boldsymbol{B}_{T,1}$ and the even rows of $\boldsymbol{B}_{T,2}$ are set to zeros. Hence, we have four types of synthetic datasets. The datasets $\{\mathcal{X}_i, y_{C,i}\}_{i=1}^n$ of a tensor size $7 \times 8 \times 5$ designated as "DataCP1" and a tensor size $9 \times 10 \times 10$ as "DataCP2" are used to validate the performance of the proposed CP-based method. The datasets $\{\mathcal{X}_i, y_{T,i}\}_{i=1}^n$ of a tensor size $7 \times 8 \times 5$ entitled "DataTucker1" and a tensor size $9 \times 10 \times 10$ as "DataTucker2" are used to evaluate the performance of the Tucker-based method. Table 1 summarizes four types of datasets we used in the simulation study.

Table 1: Types of datasets in the simulation study.

Method	Size of $\mathcal{X}$	
	$7 \times 8 \times 5$	$9 \times 10 \times 10$
CP	DataCP1	DataCP2
Tucker	DataTucker1	DataTucker2

The product quality index y_i is generated by the following rule: $y_i = 1$ if $\pi(\mathcal{X}_i) \geq 1/2$ and 0 otherwise, where $\pi(\cdot)$ is defined as below:

$$\pi(\mathcal{X}_i) = \mathbb{P}(Y_i = 1 | \mathcal{X}_i) = \frac{\exp\left[\alpha + \left\langle \mathcal{B}, \mathcal{X}_i \right\rangle\right]}{1 + \exp\left[\alpha + \left\langle \mathcal{B}, \mathcal{X}_i \right\rangle\right]}, \tag{13}$$

where we set as $\alpha = 0$, and $\mathcal{B} = \mathcal{B}_C$ or $\mathcal{B} = \mathcal{B}_T$, depending on which proposed method is chosen.

4.2 *Implementation*

We utilize four benchmark methods to demonstrate the effectiveness of the proposed methods.

(1) *Benchmark I* is an extension of Structured LASSO proposed by Zhao and Leng (2014). Structured LASSO maps the expectation of a normally distributed response variable to an explanatory matrix and decomposes the coefficient matrix as a bilinear product of two vectors. It separately penalizes the two vectors using the ℓ_1 norm to inspire two-dimensional sparsity. Structured LASSO is a two-dimensional variable selection method for applications where explanatory variables are matrices. To apply it to applications where independent variables are tensors such as the ones considered in this article, we first extend it to a logistic tensor regression model. Next, we decompose the coefficient tensor as a product of three vectors and penalize two of them using the ℓ_1 norm to induce two-dimensional sparsity. It can be easily shown that this benchmark model (i.e., the revised Structured LASSO) is a special case of the methods proposed in this article where the rank of the coefficient tensor is one for the CP-based model and $(1, 1, 1)$ for the Tucker-based model. Also, both CP- and Tucker-based models are same when their corresponding rank of a coefficient tensor is one and $(1, 1, 1)$, respectively.

(2) *Benchmark II* is a two-step sequential selection method that first identifies the informative process variables and then crucial stages. Similar to the method proposed in this article, Benchmark II is a logistic regression model with an explanatory variable of a tensor form. To achieve the goal of two-dimensional variable selection, we first penalize the Frobenius norm of each horizontal slice (a matrix) of the coefficient tensor to identify informative process variables. Next, we remove the data of these non-informative process variables from the explanatory tensor. Then, we build another logistic tensor regression model and penalize the Frobenius norm of each lateral slice (a matrix) of the coefficient tensor to select the crucial stages.

(3) *Benchmark III* is another two-step sequential selection method that is the same as Benchmark II except that it identifies the crucial stages first and then the informative process variables.

(4) *Benchmark IV* is a two-dimensional variable selection method that simultaneously identifies process variables and stages when process data is of a matrix form (Jeong and Fang, 2022). Since the benchmark requires process data to be a matrix form, we follow the procedure in Jeong and Fang (2022) that transforms each process tensor $\mathcal{X}_i \in \mathbb{R}^{P_1 \times P_2 \times P_3}$ into a matrix $\boldsymbol{X}_i \in \mathbb{R}^{P_1 \times P_2}$ by taking the average of the time-series signal of each process variable (at each stage location) and using the mean value to represent the whole signal.

We apply the benchmarks, denoted as "Benchmarks I, II, III, IV", as well as our proposed methods, denoted as "CP" and "Tucker", to the generated datasets with superimposed random noise, i.e., $\{\mathcal{X}_i + \mathcal{E}_i, y_{C,i}\}_{i=1}^n$ and $\{\mathcal{X}_i + \mathcal{E}_i, y_{T,i}\}_{i=1}^n$, respectively, where $\mathcal{E}_i$ is a random noisy tensor whose elements are randomly generated from $N(0, \sigma^2)$. In this simulation study, we experiment all the methods with multiple noise levels: $\sigma = 0.2, 0.4, 0.6$. Also, we use the logistic regression model to identify the crucial process variables and their stage locations that are responsible for product defects because the quality indices $y_{C,i}$ and $y_{T,i}$ are binary variables.

Since the optimization criterion in (6) is nonconvex, we need to try multiple initial points and choose the best model that provides the lowest loss function value. However, this process may be time-consuming. To save computation time, we propose the following heuristic initialization method. The method works by first regressing y_i against each element of $\mathcal{X}_i$ using logistic regression and constructing the regression coefficient tensor $\tilde{\mathcal{B}}$. Next, the tensor $\tilde{\mathcal{B}}$ is expanded using the CP/Tucker decomposition (depending on which method is implemented, that is, the CP- or Tucker-based method), and the results (basis/factor matrices and the core tensor) are used as the initial point of the proposed BCPD algorithm. The rank used for the CP/Tucker decomposition and tuning parameter are selected using AICc (discussed in Section 2). The algorithmic parameters for the termination condition are set as $\epsilon = 10^{-3}$ and $\delta = 500$. We use a constant step size $\tau_j = 10^{-3}$, $j = 0, \ldots, 4$ for model training.

We evaluate the performance of our proposed methods and the four benchmarks through selection accuracy and precision (or stability) of identifying the crucial process variables and their stage locations. The accuracy is calculated as the ratio of "$TP + TN$" to "the number of horizontal slices + the number of lateral slices" of the coefficient tensor, where TP represents "True Positive", which is the number of important horizontal and lateral

slices that are selected correctly, and TN means "True Negative", which is the number of non-important horizontal and lateral slices that are removed correctly. We summarize the statistics for performance evaluation as follows:

$$TPR = \frac{TP}{TP + FN}, \quad TNR = \frac{TN}{FP + TN}, \tag{14}$$

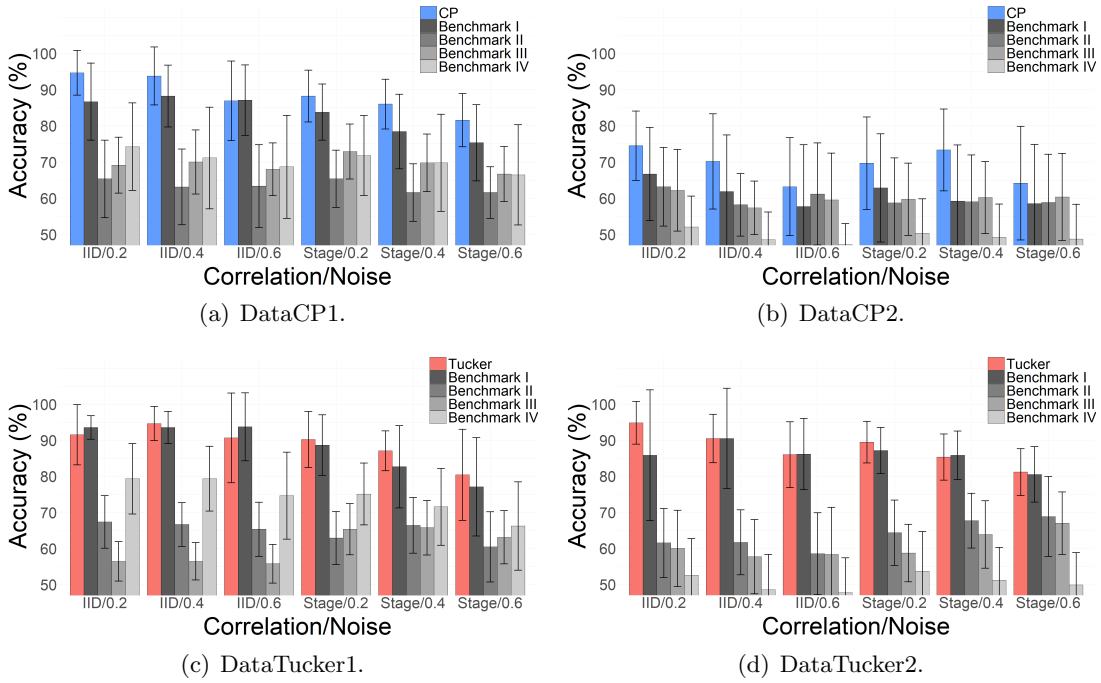
where FN is "False Negative", FP denotes "False Positive", and "R" stands for "Rate". Thus, it holds that $FNR = 1 - TPR$ and $FPR = 1 - TNR$.

We repeat the simulation experiment 30 times and report the average selection accuracy (representing diagnostic accuracy) and the standard deviation of the selection accuracy (representing diagnostic precision).

4.3 *Results and Analysis*

We summarize the diagnostic results in Tables 2–5. Tables 2–5 respectively summarize the selection accuracy and precision of each method with DataCP1, DataCP2, DataTucker1, and DataTucker2 in terms of all combinations of scenarios: sample size ($n = 200, 300, 500$), noise level ($\sigma = 0.2, 0.4, 0.6$), and correlation structure (IID and stage correlation). Figures 6, 7, and 8 depict the selection accuracy (bar plot) and precision (error bar) for each method with respect to each scenario in the case of $n = 200$, 300, 500, respectively.

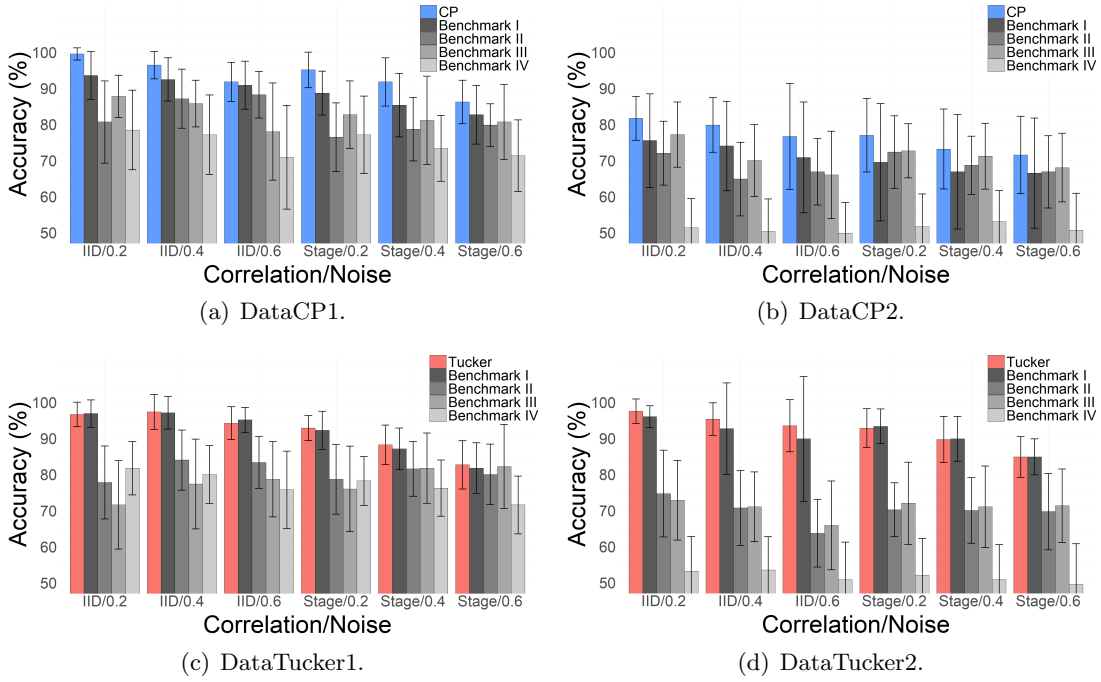
The diagnostic results indicate that the proposed CP- and Tucker-based methods achieve higher or comparable selection accuracy and precision than the benchmarks regardless of whether process data are correlated or not. For example, Tables 2, 3 and Figure 7 show that the mean accuracy and corresponding standard deviation (SD) for the CP-based method and four benchmarks are respectively 96.4 (3.8), 92.4 (6.0), 87.1 (8.2), 85.8 (6.5), 77.1 (11.0) using DataCP1, and for the Tucker-based method and four benchmarks, they are 97.3 (4.8), 97.1 (4.5), 84.0 (8.3), 77.3 (12.5), 80.0 (8.0) using DataTucker1, respectively, when there is no correlation in the process data and the noise level is set as $\sigma = 0.4$ in case of $n = 300$. Tables 2 and 3 demonstrate that the accuracy (and SD) for the CP-based method and four benchmarks are respectively 91.8 (6.7), 85.3 (8.8), 78.7 (8.8), 81.1 (12.3), 73.3 (9.1) using DataCP1, and for the Tucker-based method and four benchmarks, they are 88.2 (5.4), 87.1 (5.8), 81.6 (7.6), 81.8 (9.7), 76.2 (7.8) using DataTucker1 when there exists correlation in



(a) DataCP1. (b) DataCP2.

(c) DataTucker1. (d) DataTucker2.

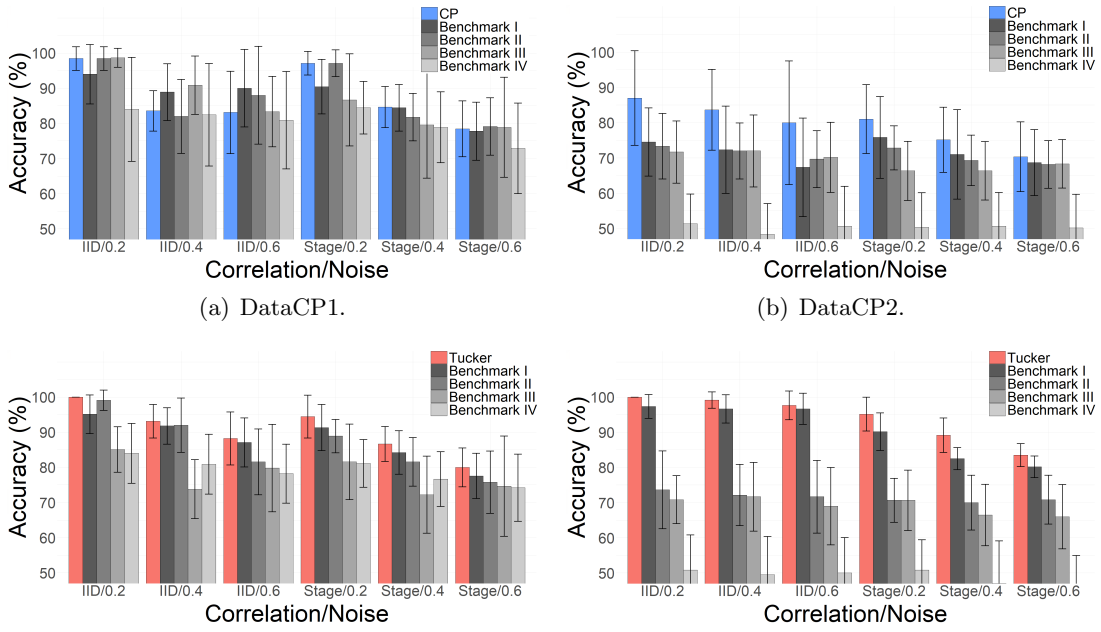
Figure 6: Results of mean accuracy (bar plot) and standard deviation (error bar) for each method using 200 samples.



(a) DataCP1. (b) DataCP2.

(c) DataTucker1. (d) DataTucker2.

Figure 7: Results of mean accuracy (bar plot) and standard deviation (error bar) for each method with 300 samples.



(a) DataCP1. (b) DataCP2.

(c) DataTucker1. (d) DataTucker2.

Figure 8: Results of mean accuracy (bar plot) and standard deviation (error bar) for each method with 500 samples.

the process data and $\sigma = 0.4$ in case of $n = 300$.

Tables 2–5 as well as Figures 6–8 suggest that our proposed methods tend to perform better than the benchmarks in terms of both selection accuracy and precision no matter what the noise level and the sample size are. We believe this is because Benchmark I does not try to find the true rank, but our proposed methods try to recover the true model with the true rank. However, the CP-based method tries a limited number of ranks, and Tucker-based method uses HOSVD rather than testing all rank combinations, so they somehow show the comparable or slightly worse results with those of Benchmark I. Also, since Benchmarks II and III select important process variables (or stages) first and then identify the important stages (or process variables) next, the first procedure negatively affects the second procedure, which makes the accuracy generally lower than the proposed methods which select important process variables and stages simultaneously. Moreover, Benchmark IV shows the low accuracy and precision because it loses some useful information when transforming the 3D tensor data into a 2D matrix, while our proposed methods directly uses the 3D tensor data without any transformation, showing the advantage of our proposed methods.

Table 2: Mean selection accuracy (%) using DataCP1 with tensors of size $(7 \times 8 \times 5)$ (values inside parantheses are standard deviations).

n	σ	Method	IID					Stage Correlation				
			TPR	TNR	FNR	FPR	Accuracy	TPR	TNR	FNR	FPR	Accuracy
200	0.2	CP	91.9 (8.9)	97.1 (6.3)	8.1 (8.9)	2.9 (6.3)	**94.7** (6.2)	84.8 (9.1)	91.3 (9.9)	15.2 (9.1)	8.8 (9.9)	88.2 (7.2)
		Ben I	76.2 (23.5)	95.8 (7.6)	23.8 (23.5)	4.2 (7.6)	86.7 (10.6)	76.7 (14.3)	90.0 (9.5)	23.3 (14.3)	10.0 (9.5)	**83.8** (7.8)
		Ben II	62.4 (15.7)	67.9 (8.5)	37.6 (15.7)	32.1 (8.5)	65.3 (10.7)	66.7 (10.2)	64.2 (10.2)	33.3 (10.2)	35.8 (10.2)	65.3 (7.9)
		Ben III	77.1 (9.6)	62.1 (10.6)	22.9 (9.6)	37.9 (10.6)	69.1 (7.7)	80.5 (7.0)	66.3 (9.9)	19.5 (7.0)	33.8 (9.9)	72.9 (7.6)
		Ben IV	63.3 (21.4)	83.8 (14.4)	36.7 (21.4)	16.3 (14.4)	74.2 (12.1)	57.6 (18.9)	84.2 (13.5)	42.4 (18.9)	15.8 (13.5)	71.8 (11)
	0.4	CP	91.9 (10.4)	95.4 (9.0)	8.1 (10.4)	4.6 (9.0)	**93.8** (8.0)	80.0 (11.6)	91.3 (9.4)	20.0 (11.6)	8.8 (9.4)	**86.0** (6.9)
		Ben I	79.0 (16.2)	96.3 (7.4)	21.0 (16.2)	3.8 (7.4)	88.2 (8.5)	66.7 (23.8)	88.8 (10.6)	33.3 (23.8)	11.3 (10.6)	78.4 (10.3)
		Ben II	60.0 (14.7)	65.8 (8.6)	40.0 (14.7)	34.2 (8.6)	63.1 (10.5)	61.9 (10.8)	61.3 (9.5)	38.1 (10.8)	38.8 (9.5)	61.6 (8.0)
		Ben III	76.2 (10.8)	64.6 (10.4)	23.8 (10.8)	35.4 (10.4)	70.0 (8.9)	76.2 (10.2)	64.2 (8.5)	23.8 (10.2)	35.8 (8.5)	69.8 (8.0)
		Ben IV	56.2 (24.6)	84.2 (13.5)	43.8 (24.6)	15.8 (13.5)	71.1 (14.0)	56.7 (22)	81.3 (13.0)	43.3 (22.0)	18.8 (13.0)	69.8 (13.4)
	0.6	CP	77.6 (20.8)	95.0 (10.2)	22.4 (20.8)	5.0 (10.2)	86.9 (11.0)	74.3 (16.9)	87.9 (9.0)	25.7 (16.9)	12.1 (9.0)	**81.6** (7.4)
		Ben I	75.7 (19.9)	97.1 (5.4)	24.3 (19.9)	2.9 (5.4)	**87.1** (9.7)	62.4 (26.4)	86.7 (12.3)	37.6 (26.4)	13.3 (12.3)	75.3 (10.5)
		Ben II	58.1 (15.9)	67.9 (9.1)	41.9 (15.9)	32.1 (9.1)	63.3 (11.4)	60.5 (9.7)	62.5 (7.3)	39.5 (9.7)	37.5 (7.3)	61.6 (7.2)
		Ben III	70.0 (11.5)	66.3 (8.8)	30.0 (11.5)	33.8 (8.8)	68.0 (7.3)	70.5 (11.2)	63.3 (8.0)	29.5 (11.2)	36.7 (8.0)	66.7 (7.6)
		Ben IV	53.3 (20.9)	82.1 (17.6)	46.7 (20.9)	17.9 (17.6)	68.7 (14.2)	54.8 (22.5)	76.7 (17.3)	45.2 (22.5)	23.3 (17.3)	66.4 (13.8)
300	0.2	CP	100.0 (0.0)	99.2 (3.2)	0.0 (0.0)	0.8 (3.2)	**99.6** (1.7)	98.1 (4.9)	92.5 (7.0)	1.9 (4.9)	7.5 (7.0)	**95.1** (4.9)
		Ben I	93.8 (8.1)	93.3 (9.1)	6.2 (8.1)	6.7 (9.1)	93.6 (6.7)	91.9 (8.1)	85.8 (7.9)	8.1 (8.1)	14.2 (7.9)	88.7 (6.1)
		Ben II	81.0 (8.7)	80.4 (23.1)	19.0 (8.7)	19.6 (23.1)	80.7 (11.4)	79.5 (10.4)	73.8 (22.3)	20.5 (10.4)	26.3 (22.3)	76.4 (9.5)
		Ben III	80.5 (8.8)	94.2 (9.7)	19.5 (8.8)	5.8 (9.7)	87.8 (5.8)	72.9 (7.8)	91.3 (15.8)	27.1 (7.8)	8.8 (15.8)	82.7 (9.4)
		Ben IV	72.9 (16.9)	83.3 (12.0)	27.1 (16.9)	16.7 (12.0)	78.4 (11.0)	71.9 (18.2)	81.7 (13.8)	28.1 (18.2)	18.3 (13.8)	77.1 (10.7)
	0.4	CP	100.0 (0.0)	93.3 (7.1)	0.0 (0.0)	6.7 (7.1)	**96.4** (3.8)	98.1 (4.9)	86.3 (10.0)	1.9 (4.9)	13.8 (10.0)	**91.8** (6.7)
		Ben I	91.4 (13.8)	93.3 (7.1)	8.6 (13.8)	6.7 (7.1)	92.4 (6.0)	91.9 (9.7)	79.6 (12.5)	8.1 (9.7)	20.4 (12.5)	85.3 (8.8)
		Ben II	79.5 (11.7)	93.8 (13.0)	20.5 (11.7)	6.3 (13.0)	87.1 (8.2)	76.7 (13.8)	80.4 (19.1)	23.3 (13.8)	19.6 (19.1)	78.7 (8.8)
		Ben III	78.6 (9.0)	92.1 (13.7)	21.4 (9.0)	7.9 (13.7)	85.8 (6.5)	71.4 (17.2)	89.6 (16.8)	28.6 (17.2)	10.4 (16.8)	81.1 (12.3)
		Ben IV	69.5 (18.7)	83.8 (11.9)	30.5 (18.7)	16.3 (11.9)	77.1 (11.0)	67.6 (15.0)	78.3 (14.7)	32.4 (15.0)	21.7 (14.7)	73.3 (9.1)
	0.6	CP	98.1 (6.2)	86.3 (11.1)	1.9 (6.2)	13.8 (11.1)	**91.8** (5.4)	94.8 (10.3)	78.8 (9.9)	5.2 (10.3)	21.3 (9.9)	**86.2** (6.0)
		Ben I	93.3 (11.1)	88.8 (11.5)	6.7 (11.1)	11.3 (11.5)	90.9 (6.7)	87.1 (13.7)	78.8 (12.3)	12.9 (13.7)	21.3 (12.3)	82.7 (8.1)
		Ben II	77.1 (12.8)	97.9 (4.7)	22.9 (12.8)	2.1 (4.7)	88.2 (6.5)	67.6 (13.0)	90.4 (9.1)	32.4 (13.0)	9.6 (9.1)	79.8 (5.9)
		Ben III	66.7 (22.3)	87.9 (21.4)	33.3 (22.3)	12.1 (21.4)	78.0 (13.5)	73.8 (14.1)	86.7 (17.7)	26.2 (14.1)	13.3 (17.7)	80.7 (10.4)
		Ben IV	63.3 (19.4)	77.5 (17.5)	36.7 (19.4)	22.5 (17.5)	70.9 (14.4)	65.7 (16.6)	76.3 (13.7)	34.3 (16.6)	23.8 (13.7)	71.3 (9.9)
500	0.2	CP	100.0 (0.0)	97.1 (6.3)	0.0 (0.0)	2.9 (6.3)	98.4 (3.4)	100.0 (0.0)	94.6 (6.3)	0.0 (0.0)	5.4 (6.3)	**97.1** (3.4)
		Ben I	99.5 (2.6)	89.2 (14.6)	0.5 (2.6)	10.8 (14.6)	94.0 (8.5)	99.0 (3.6)	82.9 (13.3)	1.0 (3.6)	17.1 (13.3)	90.4 (7.8)
		Ben II	100.0 (0.0)	97.1 (6.3)	0.0 (0.0)	2.9 (6.3)	98.4 (3.4)	100.0 (0.0)	94.6 (7.1)	0.0 (0.0)	5.4 (7.1)	**97.1** (3.8)
		Ben III	99.5 (2.6)	97.9 (4.7)	0.5 (2.6)	2.1 (4.7)	**98.7** (2.7)	89.5 (8.3)	84.2 (21.8)	10.5 (8.3)	15.8 (21.8)	86.7 (13.1)
		Ben IV	88.1 (15.0)	80.4 (20.7)	11.9 (15.0)	19.6 (20.7)	84.0 (14.8)	88.6 (12.7)	80.8 (11.2)	11.4 (12.7)	19.2 (11.2)	84.4 (7.5)
	0.4	CP	100.0 (0.0)	69.2 (10.8)	0.0 (0.0)	30.8 (10.8)	83.6 (5.7)	100.0 (0.0)	71.3 (11.0)	0.0 (0.0)	28.8 (11)	**84.7** (5.8)
		Ben I	99.5 (2.6)	79.6 (15.6)	0.5 (2.6)	20.4 (15.6)	88.9 (8.1)	98.1 (4.9)	72.5 (14.1)	1.9 (4.9)	27.5 (14.1)	84.4 (6.6)
		Ben II	100.0 (0.0)	66.3 (19.7)	0.0 (0.0)	33.8 (19.7)	82.0 (10.5)	100.0 (0.0)	65.8 (12.7)	0.0 (0.0)	34.2 (12.7)	81.8 (6.8)
		Ben III	99.0 (5.2)	83.8 (15.8)	1.0 (5.2)	16.3 (15.8)	**90.9** (8.3)	87.6 (7.2)	72.5 (26.9)	12.4 (7.2)	27.5 (26.9)	79.6 (15.2)
		Ben IV	87.1 (14.7)	78.3 (21.5)	12.9 (14.7)	21.7 (21.5)	82.4 (14.6)	81.9 (14.5)	76.3 (15.5)	18.1 (14.5)	23.8 (15.5)	78.9 (10.1)
	0.6	CP	100.0 (0.0)	68.3 (22.0)	0.0 (0.0)	31.7 (22.0)	83.1 (11.7)	100.0 (0.0)	59.6 (14.9)	0.0 (0.0)	40.4 (14.9)	78.4 (8.0)
		Ben I	99.0 (3.6)	82.1 (20.9)	1.0 (3.6)	17.9 (20.9)	**90.0** (11.0)	97.1 (5.8)	60.8 (15.3)	2.9 (5.8)	39.2 (15.3)	77.8 (8.3)
		Ben II	99.5 (2.6)	77.9 (26.4)	0.5 (2.6)	22.1 (26.4)	88.0 (13.9)	98.6 (4.4)	62.1 (16.6)	1.4 (4.4)	37.9 (16.6)	**79.1** (8.2)
		Ben III	97.1 (5.8)	71.3 (21.1)	2.9 (5.8)	28.8 (21.1)	83.3 (10.0)	85.7 (6.5)	72.9 (26.7)	14.3 (6.5)	27.1 (26.7)	78.9 (14.2)
		Ben IV	81.9 (16.7)	80.0 (18.2)	18.1 (16.7)	20.0 (18.2)	80.9 (13.9)	77.6 (14.9)	68.8 (20.4)	22.4 (14.9)	31.3 (20.4)	72.9 (12.9)

Additionally, Tables 2–5 and Figures 6–8 also indicate that both selection accuracy and precision decrease as the noise level increases. For instance, Table 3 exhibits that when there exists correlation in the process data, the mean selection accuracy (and SD) of the Tucker-based method are 92.9 (3.5), 88.2 (5.4), and 82.7 (6.7) for σ equaling 0.2, 0.4, and 0.6, respectively, using DataTucker1 in presence of stage correlation in case of $n = 300$. A similar pattern can also be observed in the CP-based method. This is reasonable since a higher noise level implies more noisy data used, and thus the selection accuracy and precision are compromised.

Furthermore, Tables 2–5 and Figures 6–8 show that the selection accuracy and precision of the proposed methods are usually worse when the process data are with correlation than the case without correlation. For example, in Table 2, when $\sigma = 0.6$ for DataCP1, the

Table 3: Mean selection accuracy (%) using DataTucker1 with tensors of size $(7 \times 8 \times 5)$ (values inside parantheses are standard deviations).

n	σ	Method	IID					Stage Correlation				
			TPR	TNR	FNR	FPR	Accuracy	TPR	TNR	FNR	FPR	Accuracy
200	0.2	Tucker	82.9 (18.5)	99.2 (3.2)	17.1 (18.5)	0.8 (3.2)	91.6 (8.4)	81 (16.1)	98.3 (4.3)	19.0 (16.1)	1.7 (4.3)	**90.2** (7.8)
		Ben I	87.6 (7.2)	98.8 (3.8)	12.4 (7.2)	1.3 (3.8)	**93.6** (3.3)	83.8 (16.7)	92.9 (8.5)	16.2 (16.7)	7.1 (8.5)	88.7 (8.4)
		Ben II	70.5 (6.4)	64.6 (11.4)	29.5 (6.4)	35.4 (11.4)	67.3 (7.3)	64.3 (9.0)	61.7 (10.9)	35.7 (9.0)	38.3 (10.9)	62.9 (7.4)
		Ben III	99.5 (2.6)	18.8 (9.7)	0.5 (2.6)	81.3 (9.7)	56.4 (5.5)	72.9 (9.5)	58.8 (7.4)	27.1 (9.5)	41.3 (7.4)	65.3 (7.1)
		Ben IV	70.0 (12.6)	87.5 (11.8)	30.0 (12.6)	12.5 (11.8)	79.3 (9.8)	64.3 (12.9)	84.6 (14.2)	35.7 (12.9)	15.4 (14.2)	75.1 (8.6)
	0.4	Tucker	89.5 (9.9)	99.2 (3.2)	10.5 (9.9)	0.8 (3.2)	**94.7** (4.8)	84.3 (6.9)	89.6 (8.7)	15.7 (6.9)	10.4 (8.7)	**87.1** (5.5)
		Ben I	88.6 (9.5)	97.9 (4.7)	11.4 (9.5)	2.1 (4.7)	93.6 (4.5)	75.2 (27)	89.2 (9.7)	24.8 (27)	10.8 (9.7)	82.7 (11.4)
		Ben II	70.5 (5.2)	63.3 (9.2)	29.5 (5.2)	36.7 (9.2)	66.7 (6.1)	66.7 (10.8)	66.3 (7.4)	33.3 (10.8)	33.8 (7.4)	66.4 (7.7)
		Ben III	98.1 (4.9)	20.0 (7.0)	1.9 (4.9)	80.0 (7.0)	56.4 (5.2)	72.4 (12.4)	60.0 (8.3)	27.6 (12.4)	40.0 (8.3)	65.8 (7.6)
		Ben IV	69.5 (12.3)	87.9 (11.6)	30.5 (12.3)	12.1 (11.6)	79.3 (9.0)	56.7 (17.0)	84.6 (15.6)	43.3 (17.0)	15.4 (15.6)	71.6 (10.6)
	0.6	Tucker	81.4 (26.8)	98.8 (3.8)	18.6 (26.8)	1.3 (3.8)	90.7 (12.5)	72.9 (30.3)	87.1 (10.1)	27.1 (30.3)	12.9 (10.1)	**80.4** (12.6)
		Ben I	89.5 (18.7)	97.5 (6.1)	10.5 (18.7)	2.5 (6.1)	**93.8** (9.4)	65.2 (34.3)	87.5 (10.4)	34.8 (34.3)	12.5 (10.4)	77.1 (13.6)
		Ben II	67.1 (9.3)	63.8 (10.0)	32.9 (9.3)	36.3 (10.0)	65.3 (7.5)	59.0 (13.4)	61.7 (9.2)	41.0 (13.4)	38.3 (9.2)	60.4 (9.7)
		Ben III	97.6 (5.4)	19.2 (7.1)	2.4 (5.4)	80.8 (7.1)	55.8 (5.4)	68.1 (15.3)	58.8 (7.4)	31.9 (15.3)	41.3 (7.4)	63.1 (7.4)
		Ben IV	64.8 (17.5)	83.3 (13.3)	35.2 (17.5)	16.7 (13.3)	74.7 (12.1)	49.5 (16.2)	80.8 (16.7)	50.5 (16.2)	19.2 (16.7)	66.2 (12.2)
300	0.2	Tucker	92.9 (7.3)	100.0 (0.0)	7.1 (7.3)	0.0 (0.0)	**96.7** (3.4)	87.1 (4.4)	97.9 (4.7)	12.9 (4.4)	2.1 (4.7)	**92.9** (3.5)
		Ben I	93.8 (7.2)	99.6 (2.3)	6.2 (7.2)	0.4 (2.3)	96.9 (3.8)	89.0 (6.1)	95.0 (7.0)	11.0 (6.1)	5.0 (7.0)	92.2 (5.3)
		Ben II	76.7 (13.3)	78.8 (23.0)	23.3 (13.3)	21.3 (23.0)	77.8 (10.1)	78.1 (9.0)	79.2 (22.6)	21.9 (9.0)	20.8 (22.6)	78.7 (9.7)
		Ben III	72.4 (3.6)	70.8 (21.6)	27.6 (3.6)	29.2 (21.6)	71.6 (12.2)	69.5 (8.2)	81.7 (22.2)	30.5 (8.2)	18.3 (22.2)	76.0 (11.8)
		Ben IV	74.3 (10.2)	88.3 (11.8)	25.7 (10.2)	11.7 (11.8)	81.8 (7.4)	70.0 (12.1)	85.4 (12.7)	30.0 (12.1)	14.6 (12.7)	78.2 (6.8)
	0.4	Tucker	96.2 (7.4)	98.3 (4.3)	3.8 (7.4)	1.7 (4.3)	**97.3** (4.8)	88.1 (5.4)	88.3 (8.0)	11.9 (5.4)	11.7 (8.0)	**88.2** (5.4)
		Ben I	96.7 (6.1)	97.5 (5.1)	3.3 (6.1)	2.5 (5.1)	97.1 (4.5)	88.6 (5.8)	85.8 (7.9)	11.4 (5.8)	14.2 (7.9)	87.1 (5.8)
		Ben II	74.3 (12.1)	92.5 (16.6)	25.7 (12.1)	7.5 (16.6)	84.0 (8.3)	69.5 (11.1)	92.1 (12.1)	30.5 (11.1)	7.9 (12.1)	81.6 (7.6)
		Ben III	72.4 (6.4)	81.7 (22.2)	27.6 (6.4)	18.3 (22.2)	77.3 (12.5)	71.4 (11.9)	90.8 (15.4)	28.6 (11.9)	9.2 (15.4)	81.8 (9.7)
		Ben IV	71.4 (9.9)	87.5 (12.7)	28.6 (9.9)	12.5 (12.7)	80.0 (8.0)	68.1 (13.4)	83.3 (13.3)	31.9 (13.4)	16.7 (13.3)	76.2 (7.8)
	0.6	Tucker	96.7 (8.1)	92.1 (7.7)	3.3 (8.1)	7.9 (7.7)	94.2 (4.5)	88.6 (6.9)	77.5 (10.6)	11.4 (6.9)	22.5 (10.6)	**82.7** (6.7)
		Ben I	98.6 (4.4)	92.1 (7.0)	1.4 (4.4)	7.9 (7.0)	**95.1** (3.5)	88.6 (8.7)	75.8 (11.8)	11.4 (8.7)	24.2 (11.8)	81.8 (7.0)
		Ben II	68.6 (15.6)	96.3 (9.9)	31.4 (15.6)	3.8 (9.9)	83.3 (7.2)	67.1 (13.1)	91.3 (11.4)	32.9 (13.1)	8.8 (11.4)	80.0 (8.4)
		Ben III	72.4 (6.4)	84.2 (19.1)	27.6 (6.4)	15.8 (19.1)	78.7 (10.4)	71.0 (17.4)	92.1 (16.2)	29.0 (17.4)	7.9 (16.2)	82.2 (11.7)
		Ben IV	69.0 (12.5)	81.7 (14.9)	31.0 (12.5)	18.3 (14.9)	75.8 (10.7)	62.4 (15.2)	79.6 (13.3)	37.6 (15.2)	20.4 (13.3)	71.6 (8.0)
500	0.2	Tucker	100.0 (0.0)	100.0 (0.0)	0.0 (0.0)	0.0 (0.0)	**100.0** (0.0)	96.7 (6.1)	92.5 (7.8)	3.3 (6.1)	7.5 (7.8)	**94.4** (6.1)
		Ben I	100.0 (0.0)	90.8 (10.3)	0.0 (0.0)	9.2 (10.3)	95.1 (5.5)	100.0 (0.0)	83.8 (12.3)	0.0 (0.0)	16.3 (12.3)	91.3 (6.6)
		Ben II	99.5 (2.6)	98.8 (3.8)	0.5 (2.6)	1.3 (3.8)	99.1 (2.9)	90.0 (6.7)	87.9 (8.4)	10.0 (6.7)	12.1 (8.4)	88.9 (4.7)
		Ben III	85.2 (2.6)	85.0 (12.5)	14.8 (2.6)	15.0 (12.5)	85.1 (6.5)	84.3 (4.4)	79.2 (17.2)	15.7 (4.4)	20.8 (17.2)	81.6 (10.7)
		Ben IV	78.1 (9.7)	89.2 (14.2)	21.9 (9.7)	10.8 (14.2)	84.0 (8.5)	73.3 (9.7)	87.9 (10.1)	26.7 (9.7)	12.1 (10.1)	81.1 (6.8)
	0.4	Tucker	100.0 (0.0)	87.1 (9.0)	0.0 (0.0)	12.9 (9.0)	**93.1** (4.8)	97.6 (5.4)	77.1 (9.3)	2.4 (5.4)	22.9 (9.3)	**86.7** (5.0)
		Ben I	100.0 (0.0)	84.6 (9.7)	0.0 (0.0)	15.4 (9.7)	91.8 (5.2)	99.5 (2.6)	70.8 (11.1)	0.5 (2.6)	29.2 (11.1)	84.2 (6.2)
		Ben II	100.0 (0.0)	85.0 (14.5)	0.0 (0.0)	15.0 (14.5)	92.0 (7.7)	94.8 (7.0)	70.0 (14.5)	5.2 (7.0)	30.0 (14.5)	81.6 (6.9)
		Ben III	86.7 (5.2)	62.5 (13.9)	13.3 (5.2)	37.5 (13.9)	73.8 (8.4)	84.8 (3.6)	61.3 (19.2)	15.2 (3.6)	38.8 (19.2)	72.2 (10.9)
		Ben IV	77.1 (10.3)	84.2 (16.1)	22.9 (10.3)	15.8 (16.1)	80.9 (8.5)	72.4 (10.6)	80.4 (14.6)	27.6 (10.6)	19.6 (14.6)	76.7 (7.8)
	0.6	Tucker	100.0 (0.0)	77.9 (14.2)	0.0 (0.0)	22.1 (14.2)	**88.2** (7.6)	93.8 (7.2)	67.9 (13.0)	6.2 (7.2)	32.1 (13.0)	**80.0** (5.5)
		Ben I	100.0 (0.0)	75.8 (13.1)	0.0 (0.0)	24.2 (13.1)	87.1 (7.0)	96.2 (6.4)	61.3 (12.0)	3.8 (6.4)	38.8 (12.0)	77.6 (6.4)
		Ben II	97.1 (7.9)	67.9 (20.7)	2.9 (7.9)	32.1 (20.7)	81.6 (9.4)	95.7 (6.7)	58.3 (19.2)	4.3 (6.7)	41.7 (19.2)	75.8 (8.8)
		Ben III	87.1 (10.2)	73.3 (22.7)	12.9 (10.2)	26.7 (22.7)	79.8 (12.4)	85.2 (2.6)	65.4 (27.2)	14.8 (2.6)	34.6 (27.2)	74.7 (14.3)
		Ben IV	73.8 (12.5)	82.1 (16.6)	26.2 (12.5)	17.9 (16.6)	78.2 (8.4)	70.5 (11.2)	77.5 (15.9)	29.5 (11.2)	22.5 (15.9)	74.2 (9.5)

mean selection accuracy (and SD) of the CP-based method is 91.8 (5.4) when there is no correlation, while it is 86.2 (6.0) when the process data have correlation in case of $n = 300$. As another example, in Table 3, when $\sigma = 0.6$ for DataTucker1, the mean selection accuracy (and SD) of the Tucker-based method is 94.2 (4.5) when there is no correlation, while it is 82.7 (6.7) when the process data are correlated. This is reasonable because the existence of correlation usually compromises the performance of statistical variable selection methods.

Lastly, Tables 2–5 and Figures 6–8 indicate that there is a clear trend that the selection accuracy of the proposed methods and other alternatives increase as sample size increases. This occurs because a larger number of samples lead to more accurate model estimation, consequently yielding higher selection accuracy. Also, when the size of tensors increases from $(7 \times 8 \times 5)$ to $(9 \times 10 \times 10)$, the selection accuracy deteriorates. Similar to the above

Table 4: Mean selection accuracy (%) using DataCP2 with tensors of size $(9 \times 10 \times 10)$ (values inside parantheses are standard deviations).

n	σ	Method	IID					Stage Correlation				
			TPR	TNR	FNR	FPR	Accuracy	TPR	TNR	FNR	FPR	Accuracy
200	0.2	CP	63.3 (11.8)	85.7 (11.4)	36.7 (11.8)	14.3 (11.4)	**74.5** (9.6)	58.0 (13.5)	81.3 (13.6)	42.0 (13.5)	18.7 (13.6)	**69.7** (12.8)
		Ben I	63.0 (14.7)	70.3 (21.1)	37.0 (14.7)	29.7 (21.1)	66.7 (12.9)	55.3 (16.3)	70.3 (17.1)	44.7 (16.3)	29.7 (17.1)	62.8 (15.0)
		Ben II	62.0 (11.0)	64.3 (17.0)	38.0 (11.0)	35.7 (17.0)	63.2 (10.9)	57.0 (9.5)	60.3 (19.0)	43.0 (9.5)	39.7 (19.0)	58.7 (12.5)
		Ben III	60.7 (9.4)	63.7 (16.9)	39.3 (9.4)	36.3 (16.9)	62.2 (11.3)	61.3 (10.1)	58.0 (13.5)	38.7 (10.1)	42.0 (13.5)	59.7 (10.0)
		Ben IV	23.0 (16.8)	81.0 (11.2)	77.0 (16.8)	19.0 (11.2)	52.0 (8.6)	16.7 (16.0)	84.0 (15.7)	83.3 (16.0)	16.0 (15.7)	50.3 (9.5)
	0.4	CP	61.0 (17.1)	79.3 (12.3)	39.0 (17.1)	20.7 (12.3)	**70.2** (13.2)	65.3 (11.4)	81.3 (18.7)	34.7 (11.4)	18.7 (18.7)	**73.3** (11.3)
		Ben I	57.0 (18.6)	66.7 (19.9)	43.0 (18.6)	33.3 (19.9)	61.8 (15.6)	56.3 (14.0)	62.0 (23.4)	43.7 (14.0)	38.0 (23.4)	59.2 (15.5)
		Ben II	57.0 (7.5)	59.3 (15.1)	43.0 (7.5)	40.7 (15.1)	58.2 (8.7)	54.3 (9.4)	63.7 (20.4)	45.7 (9.4)	36.3 (20.4)	59.0 (13.0)
		Ben III	56.0 (8.9)	58.7 (12.0)	44.0 (8.9)	41.3 (12.0)	57.3 (7.4)	58.0 (6.6)	62.3 (16.5)	42.0 (6.6)	37.7 (16.5)	60.2 (10.0)
		Ben IV	18.0 (13.5)	79.0 (16.7)	82.0 (13.5)	21.0 (16.7)	48.5 (7.7)	17.0 (12.9)	81.3 (15.0)	83.0 (12.9)	18.7 (15.0)	49.2 (9.2)
	0.6	CP	58.0 (15.4)	68.3 (17.8)	42.0 (15.4)	31.7 (17.8)	**63.2** (13.5)	56.3 (12.2)	72.0 (22.3)	43.7 (12.2)	28.0 (22.3)	**64.2** (15.7)
		Ben I	54.7 (21.1)	60.7 (18.9)	45.3 (21.1)	39.3 (18.9)	57.7 (17.1)	57.7 (17.2)	59.3 (25.9)	42.3 (17.2)	40.7 (25.9)	58.5 (16.4)
		Ben II	55.3 (15.3)	67.0 (16.8)	44.7 (15.3)	33.0 (16.8)	61.2 (14.1)	51.3 (14.3)	66.3 (16.1)	48.7 (14.3)	33.7 (16.1)	58.8 (13.3)
		Ben III	53.7 (11.3)	65.3 (17.6)	46.3 (11.3)	34.7 (17.6)	59.5 (13.0)	54.0 (13.0)	66.7 (15.8)	46.0 (13.0)	33.3 (15.8)	60.3 (12.0)
		Ben IV	18.7 (11.7)	75.7 (9.7)	81.3 (11.7)	24.3 (9.7)	47.2 (5.8)	18.7 (12.5)	78.7 (13.6)	81.3 (12.5)	21.3 (13.6)	48.7 (9.6)
300	0.2	CP	72.7 (5.8)	90.7 (7.4)	27.3 (5.8)	9.3 (7.4)	**81.7** (6.1)	68.7 (10.4)	85.3 (11.4)	31.3 (10.4)	14.7 (11.4)	**77.0** (10.2)
		Ben I	69.7 (11.9)	81.3 (17.4)	30.3 (11.9)	18.7 (17.4)	75.5 (13.0)	64.7 (16.6)	74.3 (19.1)	35.3 (16.6)	25.7 (19.1)	69.5 (16.3)
		Ben II	69.7 (8.5)	74.3 (13.0)	30.3 (8.5)	25.7 (13.0)	72.0 (8.9)	72.3 (10.1)	72.3 (13.3)	27.7 (10.1)	27.7 (13.3)	72.3 (10.1)
		Ben III	73.3 (9.2)	81.0 (12.1)	26.7 (9.2)	19.0 (12.1)	77.2 (9.1)	71.7 (7.0)	73.7 (13.3)	28.3 (7.0)	26.3 (13.3)	72.7 (7.5)
		Ben IV	21.0 (14.2)	81.7 (11.5)	79.0 (14.2)	18.3 (11.5)	51.3 (8.1)	19.7 (14.0)	83.7 (13.8)	80.3 (14.0)	16.3 (13.8)	51.7 (9.0)
	0.4	CP	71.7 (6.5)	88.0 (10.0)	28.3 (6.5)	12.0 (10.0)	**79.8** (7.6)	65.3 (9.4)	81.0 (14.0)	34.7 (9.4)	19.0 (14.0)	**73.2** (11.1)
		Ben I	71.3 (11.7)	76.7 (19.9)	28.7 (11.7)	23.3 (19.9)	74.0 (12.4)	62.0 (14.9)	71.7 (19.8)	38.0 (14.9)	28.3 (19.8)	66.8 (15.9)
		Ben II	68.3 (9.1)	61.3 (13.6)	31.7 (9.1)	38.7 (13.6)	64.8 (10.2)	68.0 (8.5)	69.3 (12.8)	32.0 (8.5)	30.7 (12.8)	68.7 (8.1)
		Ben III	68.7 (9.7)	71.3 (13.6)	31.3 (9.7)	28.7 (13.6)	70.0 (10.0)	69.3 (8.3)	73.0 (13.2)	30.7 (8.3)	27.0 (13.2)	71.2 (9.2)
		Ben IV	17.3 (14.4)	83.3 (13.2)	82.7 (14.4)	16.7 (13.2)	50.3 (9.0)	24.3 (14.5)	81.7 (14.2)	75.7 (14.5)	18.3 (14.2)	53.0 (8.7)
	0.6	CP	69.3 (13.6)	84.0 (17.1)	30.7 (13.6)	16.0 (17.1)	**76.7** (14.7)	65.0 (11.4)	78.0 (12.1)	35.0 (11.4)	22.0 (12.1)	**71.5** (10.8)
		Ben I	73.0 (12.4)	68.7 (23.0)	27.0 (12.4)	31.3 (23.0)	70.8 (15.4)	66.0 (16.1)	6.07 (20.7)	34.0 (16.1)	33.0 (20.7)	66.5 (15.3)
		Ben II	63.7 (9.6)	70.0 (12.6)	36.3 (9.6)	30.0 (12.6)	66.8 (9.2)	66.0 (9.7)	67.7 (13.8)	34.0 (9.7)	32.3 (13.8)	66.8 (10.0)
		Ben III	64.7 (12.8)	67.3 (15.5)	35.3 (12.8)	32.7 (15.5)	66.0 (12.1)	66.0 (8.1)	70.0 (13.6)	34.0 (8.1)	30.0 (13.6)	68.0 (9.5)
		Ben IV	15.3 (11.1)	84.3 (14.5)	84.7 (11.1)	15.7 (14.5)	49.8 (8.5)	21.0 (15.2)	80.3 (15.0)	79.0 (15.2)	19.7 (15.0)	50.7 (10.2)
500	0.2	CP	89.7 (11.9)	84.3 (15.7)	10.3 (11.9)	15.7 (15.7)	**87.0** (13.4)	83.7 (8.9)	78.3 (11.8)	16.3 (8.9)	21.7 (11.8)	**81.0** (9.8)
		Ben I	82.0 (8.1)	67.0 (13.9)	18.0 (8.1)	33.0 (13.9)	74.5 (9.7)	81.0 (9.9)	70.7 (15.3)	19.0 (9.9)	29.3 (15.3)	75.8 (11.6)
		Ben II	84.7 (6.8)	62.0 (13.0)	15.3 (6.8)	38.0 (13.0)	73.3 (9.3)	83.0 (4.7)	62.7 (9.4)	17.0 (4.7)	37.3 (9.4)	72.8 (6.3)
		Ben III	83.0 (6.0)	60.3 (13.5)	17.0 (6.0)	39.7 (13.5)	71.7 (8.8)	78.7 (5.7)	54.0 (11.6)	21.3 (5.7)	46.0 (11.6)	66.3 (8.4)
		Ben IV	24.3 (14.3)	78.3 (11.5)	75.7 (14.3)	21.7 (11.5)	51.3 (8.4)	20.7 (18.9)	80.0 (15.8)	79.3 (18.9)	20.0 (15.8)	50.3 (9.8)
	0.4	CP	86.0 (10.7)	81.3 (12.8)	14.0 (10.7)	18.7 (12.8)	**83.7** (11.4)	79.3 (9.1)	71.0 (12.1)	20.7 (9.1)	29.0 (12.1)	**75.2** (9.2)
		Ben I	78.7 (12.2)	66.0 (14.3)	21.3 (12.2)	34.0 (14.3)	72.3 (12.4)	77.3 (13.4)	64.7 (15.0)	22.7 (13.4)	35.3 (15.0)	71.0 (12.7)
		Ben II	82.7 (5.2)	61.3 (11.4)	17.3 (5.2)	38.7 (11.4)	72.0 (7.9)	80.0 (5.9)	58.7 (10.1)	20.0 (5.9)	41.3 (10.1)	69.3 (7.2)
		Ben III	82.3 (9.0)	61.7 (13.4)	17.7 (9.0)	38.3 (13.4)	72.0 (10.2)	76.7 (7.6)	56.0 (11.3)	23.3 (7.6)	44.0 (11.3)	66.3 (8.3)
		Ben IV	21.3 (11.1)	75.3 (10.7)	78.7 (11.1)	24.7 (10.7)	48.3 (8.7)	23.7 (16.7)	77.7 (16.1)	76.3 (16.7)	22.3 (16.1)	50.7 (9.5)
	0.6	CP	83.0 (16.6)	77.0 (19.0)	17.0 (16.6)	23.0 (19.0)	**80.0** (17.5)	75.0 (10.4)	65.7 (13.3)	25.0 (10.4)	34.3 (13.3)	**70.3** (9.9)
		Ben I	75.3 (13.3)	59.3 (16.2)	24.7 (13.3)	40.7 (16.2)	67.3 (14.0)	75.7 (9.4)	61.7 (13.7)	24.3 (9.4)	38.3 (13.7)	68.7 (9.4)
		Ben II	79.0 (7.1)	60.3 (11.0)	21.0 (7.1)	39.7 (11.0)	69.7 (8.1)	79.0 (5.5)	57.3 (10.5)	21.0 (5.5)	42.7 (10.5)	68.2 (6.8)
		Ben III	78.7 (8.2)	61.7 (13.2)	21.3 (8.2)	38.3 (13.2)	70.2 (10.0)	78.0 (5.5)	58.7 (10.1)	22.0 (5.5)	41.3 (10.1)	68.3 (6.9)
		Ben IV	25.7 (15.5)	75.7 (13.8)	74.3 (15.5)	24.3 (13.8)	50.7 (11.3)	19.7 (15.2)	80.7 (16.2)	80.3 (15.2)	19.3 (16.2)	50.2 (9.5)

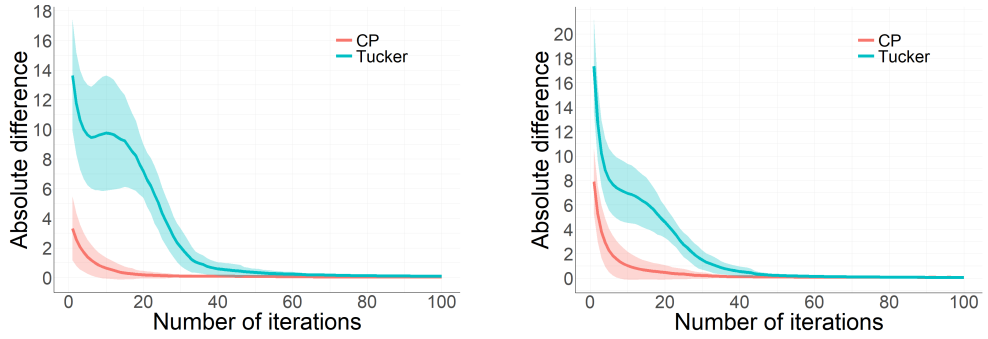
reasoning, more parameters to be estimated with the same sample sizes exacerbate model estimation and thus shows lower selection accuracy.

Table 5: Mean selection accuracy (%) using DataTucker2 with tensors of size $(9 \times 10 \times 10)$ (values inside parantheses are standard deviations).

n	σ	Method	IID					Stage Correlation				
			TPR	TNR	FNR	FPR	Accuracy	TPR	TNR	FNR	FPR	Accuracy
200	0.2	Tucker	98.0 (4.1)	91.7 (12.6)	2.0 (4.1)	8.3 (12.6)	**94.8** (5.9)	89.7 (11.0)	89.3 (7.8)	10.3 (11.0)	10.7 (7.8)	**89.5** (5.8)
		Ben I	81.7 (21.0)	90.0 (17.6)	18.3 (21.0)	10.0 (17.6)	85.8 (18.2)	85.7 (13.8)	88.7 (10.1)	14.3 (13.8)	11.3 (10.1)	87.2 (6.4)
		Ben II	63.0 (8.4)	60.0 (14.1)	37.0 (8.4)	40.0 (14.1)	61.5 (9.6)	60.7 (6.4)	68.0 (16.3)	39.3 (6.4)	32.0 (16.3)	64.3 (9.1)
		Ben III	62.3 (7.7)	57.7 (15.2)	37.7 (7.7)	42.3 (15.2)	60.0 (10.6)	58.0 (7.6)	59.3 (13.9)	42.0 (7.6)	40.7 (13.9)	58.7 (8.0)
		Ben IV	26.0 (16.7)	79.0 (14.9)	74.0 (16.7)	21.0 (14.9)	52.5 (10.2)	20.0 (16.6)	87.3 (15.7)	80.0 (16.6)	12.7 (15.7)	53.7 (11.0)
	0.4	Tucker	93.3 (8.4)	87.7 (14.8)	6.7 (8.4)	12.3 (14.8)	**90.5** (6.7)	87.7 (12.8)	83.0 (11.2)	12.3 (12.8)	17.0 (11.2)	85.3 (6.4)
		Ben I	88.0 (16.7)	93.0 (14.7)	12.0 (16.7)	7.0 (14.7)	**90.5** (13.9)	85.3 (12.5)	86.3 (11.9)	14.7 (12.5)	13.7 (11.9)	**85.8** (6.7)
		Ben II	59.3 (7.8)	64.0 (14.8)	40.7 (7.8)	36.0 (14.8)	61.7 (9.0)	59.7 (8.1)	75.7 (13.0)	40.3 (8.1)	24.3 (13.0)	67.7 (7.6)
		Ben III	59.7 (8.5)	55.7 (14.8)	40.3 (8.5)	44.3 (14.8)	57.7 (10.3)	58.0 (8.1)	69.7 (15.4)	42.0 (8.1)	30.3 (15.4)	63.8 (9.3)
		Ben IV	20.3 (16.3)	76.7 (14.0)	79.7 (16.3)	23.3 (14.0)	48.5 (9.8)	21.7 (15.1)	80.7 (12.0)	78.3 (15.1)	19.3 (12.0)	51.2 (9.1)
	0.6	Tucker	94.0 (6.2)	78.0 (18.8)	6.0 (6.2)	22.0 (18.8)	86.0 (9.1)	83.3 (12.4)	79.0 (10.9)	16.7 (12.4)	21.0 (10.9)	**81.2** (6.5)
		Ben I	82.3 (14.5)	90.0 (12.0)	17.7 (14.5)	10.0 (12.0)	**86.2** (9.9)	86.0 (11.6)	75.0 (17.2)	14.0 (11.6)	25.0 (17.2)	80.5 (7.7)
		Ben II	50.7 (10.5)	66.3 (16.7)	49.3 (10.5)	33.7 (16.7)	58.5 (11.4)	62.0 (11.0)	75.7 (15.0)	38.0 (11.0)	24.3 (15.0)	68.8 (11.1)
		Ben III	52.3 (14.1)	64.3 (15.7)	47.7 (14.1)	35.7 (15.7)	58.3 (13.0)	57.0 (9.5)	77.0 (10.9)	43.0 (9.5)	23.0 (10.9)	67.0 (8.7)
		Ben IV	17.3 (15.5)	78.0 (14.7)	82.7 (15.5)	22.0 (14.7)	47.7 (9.7)	18.3 (12.9)	81.3 (16.8)	81.7 (12.9)	18.7 (16.8)	49.8 (9.0)
300	0.2	Tucker	97.0 (5.3)	98.0 (4.8)	3.0 (5.3)	2.0 (4.8)	**97.5** (3.4)	95.3 (6.8)	90.3 (7.6)	4.7 (6.8)	9.7 (7.6)	92.8 (5.4)
		Ben I	92.7 (5.8)	99.3 (2.5)	7.3 (5.8)	0.7 (2.5)	96.0 (3.1)	94.7 (6.8)	92.0 (6.6)	5.3 (6.8)	8.0 (6.6)	**93.3** (4.8)
		Ben II	72.3 (8.2)	77.0 (18.8)	27.7 (8.2)	23.0 (18.8)	74.7 (12.0)	69.7 (5.6)	70.7 (12.0)	30.3 (5.6)	29.3 (12.0)	70.2 (7.5)
		Ben III	71.3 (9.0)	74.3 (15.0)	28.7 (9.0)	25.7 (15.0)	72.8 (11.0)	70.0 (7.4)	74.0 (17.9)	30.0 (7.4)	26.0 (17.9)	72.0 (11.4)
		Ben IV	27.0 (15.1)	79.3 (13.9)	73.0 (15.1)	20.7 (13.9)	53.2 (9.6)	22.3 (17.2)	81.7 (15.6)	77.7 (17.2)	18.3 (15.6)	52.0 (10.2)
	0.4	Tucker	95.7 (6.3)	95.0 (7.3)	4.3 (6.3)	5.0 (7.3)	**95.3** (4.5)	94.7 (7.8)	84.7 (11.1)	5.3 (7.8)	15.3 (11.1)	89.7 (6.4)
		Ben I	91.0 (10.3)	94.3 (17.0)	9.0 (10.3)	5.7 (17.0)	92.7 (12.7)	93.7 (7.2)	86.0 (10.4)	6.3 (7.2)	14.0 (10.4)	**89.8** (6.2)
		Ben II	71.0 (6.6)	70.3 (16.1)	29.0 (6.6)	29.7 (16.1)	70.7 (10.4)	67.3 (6.9)	72.7 (14.1)	32.7 (6.9)	27.3 (14.1)	70.0 (9.1)
		Ben III	70.0 (6.9)	72.0 (14.2)	30.0 (6.9)	28.0 (14.2)	71.0 (9.7)	69.3 (7.4)	72.7 (17.4)	30.7 (7.4)	27.3 (17.4)	71.0 (11.3)
		Ben IV	26.3 (11.0)	80.7 (13.9)	73.7 (11.0)	19.3 (13.9)	53.5 (9.2)	24.7 (14.8)	77.0 (14.4)	75.3 (14.8)	23.0 (14.4)	50.8 (9.7)
	0.6	Tucker	95.0 (5.7)	92.0 (12.1)	5.0 (5.7)	8.0 (12.1)	**93.5** (7.2)	93.7 (8.9)	76.0 (12.8)	6.3 (8.9)	24.0 (12.8)	**84.8** (5.6)
		Ben I	86.3 (19.2)	93.3 (16.5)	13.7 (19.2)	6.7 (16.5)	89.8 (17.3)	93.3 (5.5)	76.3 (10.0)	6.7 (5.5)	23.7 (10.0)	**84.8** (5.0)
		Ben II	64.7 (9.0)	62.7 (13.1)	35.3 (9.0)	37.3 (13.1)	63.7 (9.4)	69.0 (8.4)	70.3 (15.2)	31.0 (8.4)	29.7 (15.2)	69.7 (10.6)
		Ben III	64.3 (13.8)	67.3 (14.8)	35.7 (13.8)	32.7 (14.8)	65.8 (12.3)	69.0 (7.6)	73.7 (15.0)	31.0 (7.6)	26.3 (15.0)	71.3 (10.2)
		Ben IV	23.3 (13.2)	78.3 (13.7)	76.7 (13.2)	21.7 (13.7)	50.8 (10.3)	23.3 (12.4)	75.7 (15.5)	76.7 (12.4)	24.3 (15.5)	49.5 (11.2)
500	0.2	Tucker	100.0 (0.0)	100.0 (0.0)	0.0 (0.0)	0.0 (0.0)	**100.0** (0.0)	99.0 (3.1)	91.3 (7.8)	1.0 (3.1)	8.7 (7.8)	**95.2** (4.8)
		Ben I	100.0 (0.0)	94.7 (6.8)	0.0 (0.0)	5.3 (6.8)	97.3 (3.4)	100.0 (0.0)	80.3 (10.7)	0.0 (0.0)	19.7 (10.7)	90.2 (5.3)
		Ben II	84.0 (7.7)	63.3 (15.4)	16.0 (7.7)	36.7 (15.4)	73.7 (11.1)	82.3 (4.3)	59.0 (8.8)	17.7 (4.3)	41.0 (8.8)	70.7 (6.3)
		Ben III	82.3 (5.7)	59.3 (10.1)	17.7 (5.7)	40.7 (10.1)	70.8 (6.8)	81.7 (6.5)	59.7 (12.5)	18.3 (6.5)	40.3 (12.5)	70.7 (8.6)
		Ben IV	23.3 (15.4)	78.3 (15.6)	76.7 (15.4)	21.7 (15.6)	50.8 (10)	21.3 (14.8)	80.3 (15.4)	78.7 (14.8)	19.7 (15.4)	50.8 (8.6)
	0.4	Tucker	99.7 (1.8)	98.7 (3.5)	0.3 (1.8)	1.3 (3.5)	**99.2** (2.3)	97.3 (4.5)	81.0 (8.0)	2.7 (4.5)	19.0 (8.0)	**89.2** (4.9)
		Ben I	100.0 (0.0)	93.3 (8.0)	0.0 (0.0)	6.7 (8.0)	96.7 (4.0)	100.0 (0.0)	65.0 (6.3)	0.0 (0.0)	35.0 (6.3)	82.5 (3.1)
		Ben II	82.3 (6.3)	62.0 (12.4)	17.7 (6.3)	38.0 (12.4)	72.2 (8.7)	81.7 (4.6)	58.3 (11.5)	18.3 (4.6)	41.7 (11.5)	70.0 (7.8)
		Ben III	82.0 (8.5)	61.3 (13.1)	18.0 (8.5)	38.7 (13.1)	71.7 (9.8)	77.7 (7.3)	55.3 (11.4)	22.3 (7.3)	44.7 (11.4)	66.5 (8.7)
		Ben IV	21.0 (15.2)	78.0 (16.5)	79.0 (15.2)	22.0 (16.5)	49.5 (10.9)	25.0 (15.5)	69.0 (17.7)	75.0 (15.5)	31.0 (17.7)	47.0 (12.1)
	0.6	Tucker	99.0 (3.1)	96.3 (6.7)	1.0 (3.1)	3.7 (6.7)	**97.7** (4.1)	96.3 (4.9)	70.7 (6.4)	3.7 (4.9)	29.3 (6.4)	**83.5** (3.3)
		Ben I	99.3 (2.5)	94.0 (7.7)	0.7 (2.5)	6.0 (7.7)	96.7 (4.4)	100.0 (0.0)	60.3 (6.1)	0.0 (0.0)	39.7 (6.1)	80.2 (3.1)
		Ben II	81.3 (7.8)	62.0 (14.0)	18.7 (7.8)	38.0 (14.0)	71.7 (10.4)	81.3 (5.1)	60.3 (9.6)	18.7 (5.1)	39.7 (9.6)	70.8 (7.0)
		Ben III	78.3 (10.9)	59.7 (12.2)	21.7 (10.9)	40.3 (12.2)	69.0 (11.0)	76.3 (6.7)	55.7 (13.0)	23.7 (6.7)	44.3 (13.0)	66.0 (9.1)
		Ben IV	21.3 (14.6)	78.7 (15.0)	78.7 (14.6)	21.3 (15.0)	50.0 (10.1)	28.3 (14.6)	61.3 (19.4)	71.7 (14.6)	38.7 (19.4)	44.8 (10.1)

4.4 Convergence Performance

As described by Theorem 2, the proposed BCPD algorithm exhibits global convergence, meaning that the sequence it generates converges to a critical point in the optimization problem (6). To empirically verify the global convergence of the BCPD algorithm, we analyze the changes in the absolute value of the difference between consecutive objective function values, represented by $|\mathcal{F}^k - \mathcal{F}^{k-1}|$, throughout the iterations. The results consistently reveal a decreasing trend in these values, ultimately converging to zero in all experimental scenarios, which confirms the algorithm's global convergence. Figure 9 illustrates two examples of the convergence curves: (a) for the IID case with $\sigma = 0.2$ and (b) for the stage correlation case with $\sigma = 0.4$, using both DataCP1 and DataTucker1 datasets. The figure illustrates the mean and standard deviations of these absolute differences across 30 experiments, clearly showing the values diminishing and converging to zero.



(a) IID case with $\sigma = 0.2$.

(b) Stage correlation case with $\sigma = 0.4$.

Figure 9: Illustrations of the global convergence of the BCPD algorithm: the bold line represents the mean of absolute values of differences between consecutive objective function values against the number of iterations, and the shaded envelope depicts the standard deviations around the mean using DataCP1 and DataTucker1 over 30 experiments utilized in the simulation study.

5 Case Study: Multi-stream High-dimensional Signals from Successive Rolling Mill Stands

In this section, we validate the effectiveness of our methods using data obtained from successive rolling mill stands illustrated in Figure 1. The dataset is composed of 490 strip steel products, including 264 good quality products and 226 defective products. The

product quality index is binary, which is 1 if the product is defective and 0 otherwise. Following the suggestion from the engineers who work in this field, we focus on nine process variables: *the target speed of rollers*, *the measured speed of rollers*, *looper value*, *the target force on both side of the rollers*, *the measured force on the work side of rollers*, *the measured force on the transfer side of rollers*, *roller gap*, *looper height*, and *temperature*. At each stage, each of the process variables has a profile or time-series over $1,500$ measurement points.

5.1 *Data Preprocessing*

Among $1,500$ measurement points, we drop the first 300 points following the suggestion from engineers. The first 300 points correspond to the head of a product, which are not often used to determine if the product is defective or not since the width of the very beginning segment of the head is usually smaller than the width limit specification. As a result, we use a $9 \times 7 \times 1,200$ process variable tensor $\mathcal{X}_i$ for each product i for all $i = 1, \ldots, 490$.

The number of unknown parameters in the coefficient tensor is $75,600 = 9 \times 7 \times 1,200$, which is extremely large (even if the CP/Tucker decomposition is applied), given that there are less than 500 historical data samples for model training. Therefore, we first reduce the number of elements of the process variable tensor $\mathcal{X}_i$ by applying principle component analysis (PCA). In this case study, PCA is conducted as follows: (1) For each of the 63 process variables, construct matrices, $\boldsymbol{Z}_l \in \mathbb{R}^{490\times 1200}, \forall l = 1, \ldots, 63$, by stacking the measurement points of the process variables l from all the 490 products; (2) Scale each matrix $\boldsymbol{Z}_l$ to $[-1, 1]$ individually and then center it by subtracting each of its row by the column mean of the whole matrix. The standardized matrix is denoted as $\tilde{\boldsymbol{Z}}_l, \forall l$; (3) Apply singular value decomposition to $\tilde{\boldsymbol{Z}}_l$, i.e., $\tilde{\boldsymbol{Z}}_l = \boldsymbol{U}_l \boldsymbol{D}_l \boldsymbol{V}_l^\top, \forall l$; (4) Calculate $d_{l,p} = \sum_{j=1}^{p} d_{l,j} / \sum_{i=1}^{490} d_{l,i}$ for each $p = 1, \ldots, 490$, where $d_{l,i}$'s are the squared diagonal entries of matrix $\boldsymbol{D}_l, \forall l$; (5) Take the average of $d_{l,p}$'s for each p, i.e., as $\bar{d}_p = \sum_{l=1}^{63} d_{l,p}/63, \forall p$; (6) Choose the number of principal components p by using $\inf_p\{\bar{d}_p \geq 0.9\}$. As a result, p is chosen to be 3; (7) Calculate the PC scores $\tilde{\boldsymbol{Y}}_l \in \mathbb{R}^{490\times 3}$ as $\tilde{\boldsymbol{Y}}_l = \tilde{\boldsymbol{Z}}_l \tilde{\boldsymbol{V}}_l$, where $\tilde{\boldsymbol{V}}_l \in \mathbb{R}^{1200\times 3}$ is composed of the first three columns of $\tilde{\boldsymbol{V}}_l, \forall l$; (8) Reform the tensors $\tilde{\mathcal{X}}_i \in \mathbb{R}^{9\times 7\times 3}$ from the matrices $\{\tilde{\boldsymbol{Y}}_l \in \mathbb{R}^{490\times 3}\}_{l=1}^{63}$, where $\tilde{\mathcal{X}}_i$ serves as the predictor of the proposed diagnosis methods, $\forall i = 1, \ldots, 490$.

Table 6: Process Variable Selection Rate (%)

Process variable	CP	Tucker
Target speed	10	0
Measured speed	20	0
Looper value	**80**	**100**
Both side target force	10	10
Work side force	0	0
Transfer force	50	20
Roller gap	20	0
Looper height	30	10
Temperature	10	0

Table 7: Stage Selection Rate (%)

Method	1	**2**	3	4	5	6	**7**
CP	10	**100**	0	50	0	0	**50**
Tucker	0	**100**	0	40	0	0	**60**

5.2 *Inplementation Results*

To get a stable selection result, we randomly select 400 samples from the entire dataset to construct a sub-dataset and apply our methods to the sub-dataset to identify important process variables and their stage locations. We repeat this procedure 10 times and then compute the selection rates. Any process variables and their stage locations with a selection rate higher than 50% are considered as important variables and stages that are responsible for product defects. The selection results for process variables and stages are reported in Tables 6 and 7, respectively. The ranks selected by AICc are relatively low in this case study. Specifically, the identified ranks for the CP-based model are either 1 or 2, and they vary from $(1, 1, 1)$ to $(2, 2, 2)$ for the Tucker-based method.

Tables 6 and 7 indicate that both the CP- and Tucker-based methods select *Looper value* as a crucial process variable. The method proposed by Jeong and Fang (2022) selected three crucial process variables: *Looper value*, *Looper height*, and *Roller gap*. It can be seen that the process variable(s) selected by our proposed methods is a subset of the process variables identified by the method (Jeong and Fang, 2022). The authors in Jeong and Fang (2022) pointed out that the selection results are reasonable since *Looper value* and *Looper height* are used to control the tension of the steel strip between two stages, while *Roller*

gap is used to control the thickness of the steel strip, which also significantly affects the real-time value of *Looper value*. These three process variables are coupled and thus pose significant challenges for the closed-loop control system to adjust their values timely and correctly. Since our proposed methods identify less process variables whose inappropriate values might affect the quality of products, they provide more useful information to guide engineers to revise the feedback control algorithm in the hot rolling mill.

The crucial stage locations identified by the CP- and Tucker-based methods are *Stages 2* and *7*. It is known that Stages 1–3 of the hot rolling mill respectively have a speed reducer connected to the rollers to reduce the speed of the driven motors, whereas Stages 4–7 do not have any reducer. Therefore, the moving speed of the steel strip in Stages 4–7 is much higher than in Stages 1–3. This difference in equipment results in the use of two sets of control algorithms–one for the low-speed stages and another for the high-speed stages. It can be seen that the proposed methods select one stage from the low-speed category and another stage from the high-speed category. Similarly, the selected stages in Jeong and Fang (2022) are *Stages 3, 4*, and *6*, one stage from the low-speed category and two stages from the high-speed category. One possible explanation for the disagreement in the selection of specific stages between the two articles could be attributed to the existence of exceptionally high correlations (greater than 0.99) among the data from certain stages in both the low-speed and high-speed categories. It is worth pointing out that the methods presented in this article identify only one crucial stage in the high-speed category, which is fewer than the stages identified by Jeong and Fang (2022) in the same category. We believe this is because Jeong and Fang (2022) transforms the process variable tensor to a matrix form, which results in the loss of useful information. In contrast, the methods proposed in this paper utilize tensors to model the process data, preserving information without loss.

6 Conclusions

The root cause diagnosis of product defects often involves the joint identification of informative process variables and their stage locations, which is challenging since process data are usually three-dimensional (process variable $\times$ stage $\times$ measurement point). Most of the existing methods first transfer the 3D process data into a 2D matrix by averaging

the observations over time, one obvious limitation of which is the loss of information and thus both accuracy and stability of diagnostic results are compromised. To address this challenge, this article proposed new tensor-based diagnostic methods that simultaneously identify the important process variables and their stage locations related to product quality defects.

The proposed methods are based on penalized tensor regression, which regresses the quality index of a product against its process tensor data. The quality index can follow any distribution in the exponential family, so the proposed methods can be applied to various applications. To address the challenge of estimating a large number of unknown parameters with a relatively small amount of historical data, we decomposed an unknown coefficient tensor using the CP and Tucker decompositions, which expand it as a product of several low-dimensional matrices and a core tensor. It significantly reduces the number of parameters to be estimated and the number of historical data samples needed for estimation. Also, employing the tensor decompositions helps to remove or decrease the high correlation among process variables, stages, and measurement points, thus improving diagnostic accuracy. To estimate the parameters, we proposed an optimization algorithm with closed-form solutions and proved its convergence property.

The simulation study was implemented to validate the effectiveness of our proposed methods. The results indicated that the proposed CP- and Tucker-based methods achieved higher diagnostic accuracy and precision than the alternatives regardless of whether process data are correlated or not, and the proposed methods performed better than the benchmarks under various noise levels. A real-world dataset from successive rolling mill stands was used to evaluate the performance of the proposed methods as well. The proposed methods suggest that one crucial process variable and two stages are potentially related to the quality anomalies of steel strips. This will help engineers revise the feedback control algorithm to prevent future product quality defects.

7 Data Availability Statement

The data that support the findings of this study are available from the corresponding author, Dr. Xiaolei Fang, upon reasonable request.

Acknowledgements

The authors thank the editor, associate editor, and the anonymous referees for their comments and suggestions, which have significantly improved the quality of this article.

Notes on contributors

Cheoljoon Jeong received his B.S. degree in Industrial Engineering from Yonsei University, Korea, and Master's degree in Industrial and Systems Engineering from North Carolina State University. He is currently working toward a Ph.D. degree in the Department of Industrial and Operations Engineering, the University of Michigan. His research interests include industrial data science, quality and reliability engineering, design and analysis of computer experiments, energy and sustainability. He is a member of IISE, INFORMS, and IEEE.

Eunshin Byon is a Professor in the Department of Industrial and Operations Engineering at the University of Michigan. She received her B.S. and M.S. in Industrial and Systems Engineering from the Korea Advanced Institute of Science and Technology (KAIST) and Ph.D. in Industrial and Systems Engineering from Texas A&M University. Her research interests include data science, quality and reliability engineering, uncertainty quantification, energy and sustainability. She is a member of IISE, INFORMS, and IEEE.

Fei He received a Ph.D. degree from the University of Science and Technology Beijing (USTB), Beijing, China, in 2010. Currently, he is a full Professor with the Collaborative Innovation Center of Steel Technology, University of Science and Technology Beijing. His research interests include big data analytics, process modeling, quality measurement, and fault diagnosis.

Xiaolei Fang earned his PhD degree in Industrial Engineering from the H. Milton Stewart School of Industrial and Systems Engineering at the Georgia Institute of Technology, Atlanta, GA, USA, in 2018. He is currently an Associate Professor in the Edward P. Fitts Department of Industrial and Systems Engineering at North Carolina State University, Raleigh, NC, USA. His research interests are in industrial data analytics, focusing on High-Dimensional and Big Data applications across energy, manufacturing, and service sectors. Specifically, his work addresses analytical, computational, scalability, and privacy

challenges in developing statistical and optimization methods for analyzing vast complex data structures for real-time asset management and optimization.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control 19*(6), 716–723.

Balmashnova, E., M. Bruurmijn, R. Dissanayake, R. Duits, L. Kampmeijer, and T. van Noorden (2013). Image recognition of shape defects in hot steel rolling. In *Proceedings of the 84th European Study Group Mathematics with Industry (SWI 2012)*, pp. 22–38.

Battaglino, C., G. Ballard, and T. G. Kolda (2018). A practical randomized CP tensor decomposition. *SIAM Journal on Matrix Analysis and Applications 39*(2), 876–901.

Boyd, S. P. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press.

Burnham, K. P. and D. R. Anderson (2002). *Model Selection and Multimodel Inference: A practical information-theoretic approach* (2 ed.). Springer.

Carroll, J. D. and J.-J. Chang (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition. *Psychometrika 35*(3), 283–319.

Fang, X., K. Paynabar, and N. Gebraeel (2019). Image-based prognostics using penalized tensor regression. *Technometrics 61*(3), 369–384.

Faraway, J. J. (2014). *Linear Models with R* (2nd ed.). Chapman and Hall/CRC.

Gahrooei, M. R., K. Paynabar, M. Pacella, and J. Shi (2019). Process modeling and prediction with large number of high-dimensional variables using functional regression. *IEEE Transactions on Automation Science and Engineering 17*(2), 684–696.

Gaw, N., S. Yousefi, and M. R. Gahrooei (2022). Multimodal data fusion for systems improvement: A review. *IISE Transactions 54*(11), 1098–1116.

Gibson, I., D. W. Rosen, B. Stucker, and M. Khorasani (2021). *Additive Manufacturing Technologies*, Volume 17. Springer.

Hoff, P. D. (2015). Multilinear tensor regression for longitudinal relational data. *The Annals of Applied Statistics 9*(3), 1169.

Hong, M., X. Wang, M. Razaviyayn, and Z.-Q. Luo (2017). Iteration complexity analysis of block coordinate descent methods. *Mathematical Programming 163*, 85–114.

Hurvich, C. M. and C.-L. Tsai (1989). Regression and time series model selection in small samples. *Biometrika 76*(2), 297–307.

Jeong, C. and X. Fang (2022). Two-dimensional variable selection and its applications in the diagnostics of product quality defects. *IISE Transactions 54*(7), 619–629.

Jeong, C., Z. Xu, A. S. Berahas, E. Byon, and K. Cetin (2023). Multiblock parameter calibration in computer models. *INFORMS Journal on Data Science 2*(2), 116–137.

Kolda, T. G. and B. W. Bader (2009). Tensor decompositions and applications. *SIAM Review 51*(3), 455–500.

Kutner, M. H., C. J. Nachtsheim, J. Neter, and W. Li (2005). *Applied Linear Statistical Models* (5th ed.). McGraw-Hill.

McCullagh, P. and J. A. Nelder (1989). *Generalized linear models* (2 ed.). Chapman & Hall/CRC.

Meier, L., S. Van De Geer, and P. Bühlmann (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*(1), 53–71.

Miao, H., A. Wang, B. Li, and J. Shi (2022). Structural tensor-on-tensor regression with interaction effects and its application to a hot rolling process. *Journal of Quality Technology 54*(5), 547–560.

Nocedal, J. and J. Wright (2006). *Numerical Optimization* (2nd ed.). Springer.

Oseledets, I. V. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing 33*(5), 2295–2317.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 461–464.

Shen, B., R. Wang, A. C. C. Law, R. Kamath, H. Choo, and Z. J. Kong (2022). Super resolution for multi-sources image stream data using smooth and sparse tensor completion and its applications in data acquisition of additive manufacturing. *Technometrics 64*(1), 2–17.

Shen, B., W. Xie, and Z. J. Kong (2022). Smooth robust tensor completion for background/foreground separation with missing pixels: Novel algorithm with convergence guarantee. *Journal of Machine Learning Research 23*(217), 1–40.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological) 58*(1), 267–288.

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika 31*(3), 279–311.

Wang, F., M. R. Gahrooei, Z. Zhong, T. Tang, and J. Shi (2021). An augmented regression model for tensors with missing values. *IEEE Transactions on Automation Science and Engineering 19*(4), 2968–2984.

Yuan, M. and Y. Lin (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(1), 49–67.

Yue, X., J. G. Park, Z. Liang, and J. Shi (2020). Tensor mixed effects model with application to nanomanufacturing inspection. *Technometrics 62*(1), 116–129.

Zhao, J. and C. Leng (2014). Structured lasso for regression with matrix covariates. *Statistica Sinica 24*, 799–814.

Zhao, J., L. Niu, and S. Zhan (2017). Trace regression model with simultaneously low rank and row (column) sparse parameter. *Computational Statistics & Data Analysis 116*, 1–18.

Zhao, M., M. R. Gahrooei, and N. Gaw (2023). Robust coupled tensor decomposition and feature extraction for multimodal medical data. *IISE Transactions on Healthcare Systems Engineering 13*(2), 117–131.

Zhou, C. and X. Fang (2023). A supervised tensor dimension reduction-based prognostic model for applications with incomplete imaging data. *INFORMS Journal on Data Science*.

Zhou, C., Y. Su, T. Xia, and X. Fang (2023). Federated multilinear principal component analysis with applications in prognostics. *arXiv preprint arXiv:2312.06050*.

Zhou, H., L. Li, and H. Zhu (2013). Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association 108*(502), 540–552.