Family Learning: A Process Modeling Method for Cyber-Additive Manufacturing Network

Lening Wang[1], Xiaoyu Chen[1], Daniel Henkel[2], and Ran Jin (jran5@vt.edu)[1*]

[1]Grado Department of Industrial and Systems Engineering, Virginia Tech, USA

[2] Commonwealth Center for Advanced Manufacturing, USA

A cyber-additive manufacturing network (CAMNet) integrates connected additive manufacturing processes with advanced data analytics as computation services to support personalized product realization. However, highly personalized product designs (e.g., geometries) in CAMNet limit the sample size for each design, which may lead to unsatisfactory accuracy for computation services, e.g., a low prediction accuracy for quality modeling. Motivated by the modeling challenge, we proposed a data-driven model called *family learning* to jointly model similar-but-non-identical products as family members by quantifying the shared information among these products in the CAMNet. Specifically, the amount of shared information for each product is estimated by optimizing a similarity generation model based on design factors, which directly improve the prediction accuracy for the family learning model. The advantages of the proposed method are illustrated by both simulations and a real case study of the selective laser melting process. This family learning method can be broadly applied to data-driven modeling in a network with similar-but-non-identical connected systems.

Keywords: cyber-manufacturing system; quality modeling; selective laser melting

1. Introduction

A cyber-manufacturing system (CMS) associates interconnected manufacturing facilities with computation resources (e.g., fog computing and cloud computing units) to support efficient quality modeling, monitoring, diagnosis, control and decision-making (e.g., variation modeling and cost optimization) in smart manufacturing (Lee, et al. 2016, Yang, et al. 2019, Hu 2013). By embracing the CMS with similar-but-non-identical additive manufacturing (AM) processes, a cyber-additive manufacturing network (CAMNet) is proposed to efficiently realize the personalized products via advanced computation services in the CMS which can help to assign the task to the most eligible machines, modeling and controlling AM processes, and etc. (Chen, et al. 2018).

Taking a selective laser melting (SLM) process as an example in a CAMNet, Figure 1 shows a schematic of a SLM process, which is a metal powder bed fusion technique and has been widely used in automotive manufacturing, aerospace manufacturing, and medical manufacturing (Zhang, et al. 2019, Seabra, et al. 2016, Wei, et al. 2015). By adapting the product and process design according to a customer's demand, a SLM process can build one or multiple highly personalized products in complex shapes with unprecedented materials during one building process (Gibson, et al. 2010) to meet the specifications. Although the SLM process in a CAMNet is efficient to satisfy personalized needs, it has not been widely deployed in manufacturing industries. A major reason is that both the product integrity and quality defects, such as the geometric deviation, lack of fusion, or voids, have not been effectively controlled during the process (Frazier 2014).

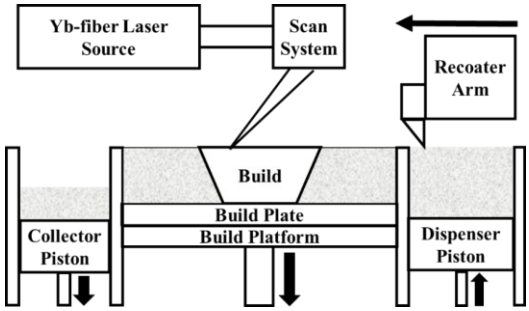

Figure 1. A schematic for a SLM process (Redrawn from (Wang et al. 2017) with authors' permission).

There are many existing data-driven methods to model the SLM processes (Strano, et al. 2013, Spears and Gold 2016). However, highly personalized products in the CAMNet pose significant challenges for modeling (Chen 2018). Specifically, the prediction performance of existing data-driven modeling methods may be poor with limited samples in personalized manufacturing. For example, aggregating data from similar-but-non-identical products without considering the heterogeneities among products may lead to unsatisfactory modeling performance, due to the unexplained variance introduced by these heterogeneities. Moreover, if we model the individual product as one model for each, even with the advanced variable selection method, such as Lasso regression (Tibshirani, 1996), if the sample size is too small, then there will not be enough degrees of freedom to support model estimation with an accurate result. Therefore, how to quantitatively measure the similarity among these similar-but-non-identical products, and further utilize the similarity measurement to improve the model performance with a limited sample size remains an open question.

In this paper, the objective is to model the quality defects (i.e., geometric deviation) caused by the fluctuations in laser during the process based on the limited samples per product. Specifically, a new model called *family learning* is proposed to model the relationship among process setting variables, *in situ* process variables, and quality variables for quality prediction. Process setting variables are the system control variables which determine the recipe of products during the fabrication process, such as laser power (online adjustable variable), scanning speed (online adjustable variable), and hatch distance (offline setting variables) in a SLM process (Olakanmi, et al. 2015). Specifically, in this study, we considered these setting variables are offline setting variables. The *in situ* process variables are collected during the fabrication process via the sensing system, such as the laser intensity and thermal distributions. The quality variables are quality measurements for the product after the fabrication process, such as the geometric deviation. The proposed family learning method focuses on providing layer-to-layer quality modeling and prediction for similar-but-non-identical products in SLM processes before post-processing (e.g., stress-relieving, product removal, etc.). In order to improve the model accuracy for these similar-but-non-identical products with limited sample size, the process setting variables, and the scanning path from each product are used to quantify the similarity among products. Moreover, by effectively quantify the similarity among multiple products, we can transfer information among products with the intuition that similar products should result in similar model coefficients. In the proposed family learning method, we employ probability mass function (pmf) to represent the similarity between these products. For example, traditional data-driven modeling methods usually fix the distribution of the model coefficients among products, such as multi-task learning (MTL) (Evgeniou and Pontil 2004) which considers a discrete uniform distribution among model coefficients. Instead of providing a fixed distribution or prior information, the proposed method initially estimates the pmf of model coefficients based on the product and process design information among products, and further updated the pmf according to the training dataset. In this way, the pmf (i.e., similarity measurements) can indeed reflect the real nonlinear similarity structure among model coefficients and further improve the model accuracy. In addition, according to the similarity measurement, for a brand-new product, we can use the model coefficients from the historical product which has the most similar product and process design to implement modeling efforts with acceptable performance.

To evaluate the performance of the proposed method, a hybrid CAMNet testbed is constructed with one SLM machine, five fused deposition machines, and 114 virtual machines which are generated from a group of linear quality-process models (Chen, et al. 2016, Wang, et al. 2018) to simulate the similar-but-non-identical processes based on the physical experiment data (Chen 2018). A simulation study is proposed to generate similar-but-non-identical products and processes based on the data from the CAMNet testbed. These similar-but-non-identical products are used to validate and compare the prediction accuracy of the proposed method and benchmark methods in a CAMNet with multiple processes. A real case study based on a fractional factorial design of experiments in a SLM process with two different product designs and multiple treatments is implemented to validate the proposed family learning. Both prediction performance and variable selection results outperform three benchmark methods, i.e., Lasso regression (Tibshirani 1996), data shared Lasso (Gross and Tibshirani 2016) and MTL (Evgeniou and Pontil 2004).

The rest part of the paper is organized as follows. The state-of-the-art modeling methods in a CMS, and data-driven models for AM processes are reviewed in Section 2. Section 3 introduces the proposed family learning method in detail. The simulation study to validate the proposed method is discussed in Section 4. A case study for a SLM process is presented in Section 5. Conclusions are drawn and future work is discussed in Section 6.

2. State-of-the-art

2.1. Modeling in a CMS

In literature, various modeling methods have been proposed for quality modeling (Wang, et al. 2010), monitoring (Jin and Liu 2013), diagnosis (Seshadrinath, et al. 2013), and process control (Huang, et al. 2014). However, in a CMS, the data from each individual product is not sufficient to support these methods in achieving satisfactory performance since the products are highly personalized. Therefore, modeling individual processes and transferring the knowledge among these processes in a CMS is an important yet challenging problem (Lee, Bagheri and Jin 2016, Zwolenski and Weatherill 2014). In literature, efforts have been made to model the heterogeneous products and processes in a CMS. For example, MTL quantifies the relationships among tasks (i.e., different designs and processes) and can lead to better performance for each task (Evgeniou and Pontil 2004). It jointly estimates the models for

all tasks simultaneously to exploit commonalities and differences among all tasks. The hierarchical Gaussian process MTL was proposed for non-parametric function learning (Li and Chen 2017). However, they do not quantitatively identify the similarity among products. On the other hand, transfer learning can help to transfer the knowledge from one domain to another domain, where domains may have different data distributions. However, adequate samples from source and target domains are required to yield an accurate model for the target domain (Pan and Yang 2010). Data shared Lasso can also borrow the information from different tasks, and isolate shared information and individual differences by using extra groups of model coefficients (Gross and Tibshirani 2016). However, data shared Lasso does not consider the interaction among these groups, and the computation is intensive compared with other methods due to the high dimensionality of the reorganized covariate matrix.

2.2. Data-driven models in the AM

For models in AM processes, several data-driven models have been proposed to identify the anomaly and control the process. Rao et al. presented an advanced Bayesian nonparametric analysis method for *in situ* sensing data (Rao, et al. 2015). It identified failures and the types of failures in a fused filament fabrication (FFF) process. This failure detection system was able to identify real-time failures. Khanzadeh et al. proposed a statistical process control strategy to detect process change via thermal images through multilinear principal component analysis (Khanzadeh, et al. 2018). Icten et al. presented a surrogate model based on polynomial chaos expansion to relate the important process parameters with product morphology. A control strategy was proposed based on the model to mitigate product variation (Icten, et al. 2015). Xing et al. presented a closed-loop control system to predict the product quality for metal powder laser forming via an industrial CCD camera and infrared photodetector (Xing, et al. 2006). The input energy on the powder bed can be predicted based on the size of the melting pool, and it can be controlled according to the prediction results during the process. Krauss et al. presented a layer-to-layer sensing system to identify the temperature distribution on the build plate. It can detect the defect areas and powder ejection during the process through temperature diffusivity analysis (Krauss, et al. 2015). However, these existed methods failed to consider similarity information among different products. Therefore, their performance may not be satisfied to model similar-but-non-identical products, especially for highly personalized products with limited sample size.

Besides, research has been reported for variation analysis and quality prediction in AM processes. For example, Sun et al. proposed a functional quantitative and qualitative model to predict two types of quality responses via offline setting variables and *in situ* process variables (Sun, et al. 2017). Shevchik et al. introduced a convolutional neural network classification model to predict the final product quality via acoustic emission signals (Shevchik, et al. 2018). Yadroitsev et al. illustrated an online optical system for the SLM process via a CCD camera (Yadroitsev, et al. 2014). The correlation between melting pool features and the microstructure of the product is identified to predict the final quality of the product. Moreover, a series of deep learning models have been investigated to predict product deviations (Huang, et al. 2020, Jin, et al. 2020, Ferreira, et al. 2019). Based on the prediction of deviation for specific CAD design, the optimal compensation plan can be implemented to improve the product geometry accuracy in AM. However, the aforementioned methods typically focus on run-to-run quality-process modeling which requires sufficient samples to estimate the model coefficients. Thus, they may not satisfy the needs of personalization in a CAMNet. On the other hand, Sabbaghi et al. proposed a series of transfer learning based models to efficient predict the geometric deviation for a new product design based on limited deviation profiles from other products (Sabbaghi and Huang 2018, Francis, et al. 2020, Sabbaghi, et al. 2018). Cheng et al. developed a statistical parametric transfer learning framework to predict the deviation profile among different designs (Cheng, et al. 2017, Cheng, et al. 2020). However, instead of learning the quality-process relationship via *in situ* process variables, they transferred the knowledge of deviation profile for each product across different process settings (e.g., size of production, material types, etc.). Besides, the transfer learning might lead to the negative transfer of knowledge when commonalities among products cannot be captured by the lurking variables (Kontar, et al. 2020).

2.3. Physics-based in the AM

On the other hand, physics-based AM models, such as finite element analysis (FEA), have been widely adopted in AM fields. For example, the purely physics-based FEA model is proposed to predict the mechanical property (i.e., elastic response) of the product (Bhandari and Lopez-Anido 2018), and also the residual distortion and stress distribution from an AM process (Chen, et al. 2019). On the other hand, in order to improve the accuracy of the FEA, hybrid models which combine the data-driven model

with the FEA have been proposed. For example, Olleak and Xi presented a modeling framework to integrate the FEA simulation with the data-driven model to improve the accuracy and efficiency of the simulation based on the limited experiment data (Olleak and Xi 2020). Li et al. proposed a FEA simulation framework with a non-parametric surrogate model to jointly estimate the thermal distribution of the AM process (Li, et al. 2018). Wang et al. proposed a meta-modeling framework to predict the high-fidelity FEA simulation results based on the corresponding low-fidelity simulation results in AM (Wang, et al. 2020). However, most of the physics-based methods mainly focused on the off-line run-to-run study for product and process designs. Due to the computation intensity, the aforementioned FEA based methods are restricted to be deployed in real fabrication stage to predict the quality measurements in real-time.

3. Methodology

3.1. Family learning for layer-to-layer modeling in the CAMNet

Without loss of generality, in this research, the geometric deviation of Product i from Layer l is treated as the quality variable in modeling, denoted as $y_{i,l}$. Moreover, to reduce the data registration complexity and computational intensity of model estimation (i.e., as described in Wang, et al. (2020)), the overall geometric deviation on each layer of a product is used to comprehensively measure the geometric difference between the design and the real product. For example, for cylinder products, the deviation is defined as the radius difference between the designed Stereolithographic (STL) file and the final products. Other quality variables can also be modeled using the same modeling method. The assumptions for the proposed family learning model include (1) the manufacturability of product which studied in this research has been validated; (2) the process similarity can be partially reflected by the designs, process settings and process characteristics (i.e., scanning patterns of products in a SLM process); (3) underlying models for similar manufacturing processes and products will have similar model structures and coefficients; in addition, there is a clustering pattern on features of product designs and manufacturing process settings, such that one can borrow information of the products from the same cluster with sufficient sample size; and (4) a linear regression model is adequate to model the quality-process relationship, which will be validated by the case study. The linear regression model for the geometric deviation $y_{i,l}$ for Layer l of Product i is formulated as:

$$y_{i,l} = \boldsymbol{w}_{i,l}^{\mathrm{T}}\boldsymbol{\beta}_i + \epsilon_{i,j}, \quad (1)$$

where $\boldsymbol{w}_{i,l}^{\mathrm{T}} \in \mathbb{R}^p$ is a vector of signal features extracted from the photodiode sensor for Product i at Layer l; p is the total number of predictors; $\boldsymbol{B} \in \mathbb{R}^{N\times p}$ is the model coefficients for N products; $\boldsymbol{\beta}_i \in \mathbb{R}^p$ represents the i-th column of $\boldsymbol{B}$ for Product i; and $\epsilon_{i,j}$ is independently and identically distributed and follows a normal distribution with zero as the mean value and constant variance. In order to estimate the model coefficients, the penalized least-square estimator can be formulated as:

$$\widehat{\boldsymbol{\beta}}_i = \underset{\boldsymbol{B},\boldsymbol{\gamma}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N}\sum_{l=1}^{L_i}\left(y_{i,l} - \boldsymbol{w}_{i,l}^{\mathrm{T}}\boldsymbol{\beta}_i\right)^2 + \rho_1\|\boldsymbol{B}\|_1 + \rho_2\sum_{i=1}^{N}\left\|\boldsymbol{\beta}_i - \sum_{j=1}^{N}\sigma(\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{m}_{ij})\boldsymbol{\beta}_j\right\|_2^2 \right\}, \quad (2)$$

where the first term $\sum_{i=1}^{N}\sum_{l=1}^{L_i}\left(y_{i,l} - \boldsymbol{w}_{i,l}^{\mathrm{T}}\boldsymbol{\beta}_i\right)^2$ represents a least-squares loss for model estimation; ρ_1 is the tuning parameter which controls the sparsity of the model; the first penalty $\|\boldsymbol{B}\|_1 = \sum_i|\boldsymbol{\beta}_i|$ is a LASSO regularization (Tibshirani 1996) term which forces the coefficients of insignificant variables to be zeros; ρ_2 is the tuning parameter that determines the importance of information borrowed from other products; in the second penalty $\sum_{i=1}^{N}\left\|\boldsymbol{\beta}_i - \sum_{j=1}^{N}\sigma(\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{m}_{ij})\boldsymbol{\beta}_j\right\|_2^2$, the difference of model coefficients between Product i and the weighted average of coefficients for all N products is minimized in order to gain more information shared from the models of the rest products; $s_{ij} = \sigma(\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{m}_{ij})$ is the similarity coefficient between Product i and j, where $\sigma(\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{m}_{ij}) = \frac{\exp(\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{m}_{ij})}{\sum_{i=1}^{N}\exp(\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{m}_{ij})}$ is the Softmax function (Bishop 2006); $\boldsymbol{\gamma}$ is a vector of weights; $\boldsymbol{m}_{ij}$ is the process and product design feature vector, which will be discussed later.

The proposed model is expected to provide satisfactory modeling accuracy given limited samples, since it shares information among similar-but-non-identical products under the assumption that their model coefficients should be also similar to each other (i.e., validated in Section 5.2). Specifically, after the transformation via the Softmax function, the range of similarity coefficient is also transferred from 0 to 1, with the summation of the coefficient is 1. Benefit from these statistical characteristics for s_{ij} (i.e., $s_{ij} \in (0,1), \sum_j s_{ij} = 1$), the process to estimate the similarity coefficient s_{ij} can be further considered as learning the probability mass function (pmf) for each system from the data

with model coefficients $\boldsymbol{\beta}_j$ as random variables. This idea can also adopt to the traditional data-driven modeling, such as MTL (Evgeniou and Pontil 2004) which considers a discrete uniform distribution for s_{ij} since it fixes the similarity measurement among different system. Therefore, instead of providing a fixed distribution or prior information, the proposed method initially estimates the pmf of model coefficients based on the product and process design information among products, and further updating the pmf according to the training dataset. In this way, the pmf (i.e., similarity measurements) can indeed reflect the real nonlinear similarity structure among model coefficients and further improve the model accuracy. To estimate the similarity measurement, we firstly denote $\boldsymbol{m}_i \in \mathbb{R}^{v_1+v_2}$ as the manufacturing feature vector, which consists of the process setting variables and summary statistics of the scanning path for product i. Then, the interaction between Product i and j is defined as $\boldsymbol{m}_{ij} = [\boldsymbol{m}_i^{\mathrm{T}}, \boldsymbol{m}_j^{\mathrm{T}}, |\boldsymbol{m}_i - \boldsymbol{m}_j|^{\mathrm{T}}]$. $\boldsymbol{m}_{ij}$ is combined by manufacturing feature vectors from product i, product j, and the absolute value of the difference between these two vectors. $\boldsymbol{m}_{ij}$ is used to estimate the similarity coefficient s_{ij} between product i and j as $s_{ij} = \sigma(\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{m}_{ij})$. The scanning path is extracted from the laser trajectory for each product, which quantifies the product design information. v_1 is the number of features getting from the process setting variables, and v_2 is the number of summary statistics getting from the functional laser trajectory on the build plate according to the Cartesian coordinate system. The summary statistics including the length, mean value, standard deviation, count of change in direction, mean value and standard deviation of the first derivative for both 2-D directions in the Cartesian coordinate system. These summaries of statistics can directly or indirectly describe the pattern of laser trajectory for each product. For example, the mean value can roughly reflect the centroid location of the product on the build area. The standard deviation and count of change in direction can represent the complexity of the laser trajectory (i.e., zig-zag trajectory). The first derivative can generally reflect the rate of change of the laser trajectory, which can reflect the turning angle and smoothness of the trajectory.

In order to efficiently estimate the model coefficient in Eq. (2), motivated by *block relaxation algorithm* (De Leeuw 1994, Lange 2010), a block updating algorithm is developed to break down the proposed optimization problem into two simpler optimization problems (see Algorithm 1). By defining $\boldsymbol{B}$ and $\boldsymbol{\gamma}$ as two variable blocks, an alternately updating strategy is employed to find the solution of the

proposed model as shown in Algorithm 1.

Algorithm 1 Block updating algorithm for minimizing Eq. (2).

Initialize: $\boldsymbol{B}^{(0)} \in \mathbb{R}^{N\times p}$ and $\boldsymbol{\gamma}^{(0)} \in \mathbb{R}^{(v_1+v_2)\times 1}$

repeat

do

$$\boldsymbol{\gamma}^{(t+1)} = \underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \left\{ \rho_2 \sum_{i=1}^{N} \left\| \boldsymbol{\beta}_i^{(t)} - \sum_{j=1}^{N} \sigma(\boldsymbol{\gamma}^{\mathrm{T}} \boldsymbol{m}_{ij}) \boldsymbol{\beta}_j^{(t)} \right\|_2^2 \right\},$$

$$\boldsymbol{B}^{(t+1)} = \underset{\boldsymbol{B}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \sum_{l=1}^{L_i} \left(\boldsymbol{y}_{i,l} - \boldsymbol{w}_{i,l}^{\mathrm{T}} \boldsymbol{\beta}_i \right)^2 + \rho_1 \|\boldsymbol{B}\|_1 + \rho_2 \sum_{i=1}^{N} \left\| \boldsymbol{\beta}_i - \sum_{j=1}^{N} \sigma(\boldsymbol{\gamma}^{(t+1)^{\mathrm{T}}} \boldsymbol{m}_{ij}) \boldsymbol{\beta}_i \right\|_2^2 \right\},$$

until $\left| \sqrt{\frac{\sum_{i=1}^{N} \sum_{l=1}^{L_i} \left(y_{i,l} - \boldsymbol{w}_{i,l}^{\mathrm{T}} \boldsymbol{\beta}_i^{(t)} \right)^2}{\sum_{i=1}^{N} L_i}} - \sqrt{\frac{\sum_{i=1}^{N} \sum_{l=1}^{L_i} \left(y_{i,l} - \boldsymbol{w}_{i,l}^{\mathrm{T}} \boldsymbol{\beta}_i^{(t+1)} \right)^2}{\sum_{i=1}^{N} L_i}} \right| \Big/ \left| \sqrt{\frac{\sum_{i=1}^{N} \sum_{l=1}^{L_i} \left(y_{i,l} - \boldsymbol{w}_{i,l}^{\mathrm{T}} \boldsymbol{\beta}_i^{(t)} \right)^2}{\sum_{i=1}^{N} L_i}} \right| < tol$

For $\boldsymbol{\gamma}^{(t)}$, the optimization problem can be efficiently solved by interior-point method (Forsgren, et al. 2002, Mehrotra 1992). Interior-point method is an efficient optimization method for both linear and nonlinear problems. The optimization problem for $\boldsymbol{B}$ is a convex problem (Zhou, et al. 2011), and can be efficiently solved via accelerated gradient descent (Chen, et al. 2009, Nesterov 2008). By adding the Nesterov's momentum term, the accelerated gradient descent can balance the gradient updates and proper extrapolation for optimization with nearly the same cost of ordinary gradient descent. To select the optimal tuning parameters ρ_1 and ρ_2, the 5-fold cross-validation is employed. In 5-fold cross-validation, firstly, the whole training dataset is randomly separated into five subsets with the almost equal number of samples. Then, four out of five subsets will be used for training, and the remaining one will be used for validation of the model. The cross-validation process is then repeated five times, with each of the five subsets used exactly once as the validation set. Based on the performance of the model under different tuning parameters, the best combination of tuning parameters can be selected (Tibshirani, et al. 2005). The root-mean-squared errors (RMSEs) from the cross-validation is used to select two tuning parameters.

4. Simulation

4.1. Simulation setup

Because the physical experiment to physically quantify the quality performance of a SLM product is usually time-consuming and economic expensive, a simulation study is employed (Montgomery 2017). The objective of this simulation study is to evaluate the statistical performance of the proposed model comparing with other benchmark models under the assumptions introduced in the Methodology section. Therefore, in the simulation study, suppose the underlying quality-process model is $y_{i,l} = \boldsymbol{w}_{i,l}^{\mathrm{T}}\boldsymbol{\beta}_{i,l} + k \cdot \varepsilon$, where $k = \frac{Var(\boldsymbol{w}_{i,l}^{\mathrm{T}}\boldsymbol{\beta}_{i,l})}{rVar(\varepsilon)}$, r is the signal-to-noise ratio, ε is the error term and follows a standard normal distribution $N(0,1)$, and other parameters follow the same definitions in Eq.(1).

Table 1. Simulation Settings.

Simulation Cases	Case 1	Case 2	Case 3	Case 4
No. of Products	10	10	100	100
No. of Layers	20	50	20	50
No. of Clusters	3	10	3	10
Signal-to-noise Ratio	10	20	10	20

In the simulation, there are four cases which are shown in Table 1. The number of products represents how many products are fabricated in each case. The number of layers is the average number of layers of these products. From the definition of the similarity coefficient in Methodology, it can be known that if some of the products have similar product designs and process settings (i.e., clustering pattern), these products, which are in the same cluster, can share more information among each other and tend to have similar model coefficients. Therefore, the number of clusters in Table 1 represents how many clusters are existed in each simulation case. For the products in the same cluster, their manufacturing feature vectors also tend to be similar to each other compared with products that are not in the cluster. The signal-to-noise ratio indicates the numerical value of r in the simulation, which will be introduced later in details.

For each simulation case, first, the number of layers for each simulated product is generated. Assume that the total layer number for product n is L_n, which follows a discrete uniform distribution

$U\{(Avg-5),(Avg+5)\}$, where Avg is the "No. of Layers" shown in Table 1. In each simulation case, the products are randomly assigned into different clusters with random permutation following a uniform distribution. The clusters determine the manufacturing feature vectors $\boldsymbol{m}$, which represent the information of designs and settings. Therefore, for product i, the manufacturing feature vector $\boldsymbol{m}_i$ can be generated from a normal distribution: $\boldsymbol{m}_i \sim N(c_i, 0.01)$, where $\boldsymbol{m}_i$ is the manufacturing feature vector for Product i; c_i is the mean value of the distribution, which is determined by the numerical order of cluster for product i. For example, in simulation Case 1, if product 1, product 2 and product 3 are from different clusters, then $\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3$ are generated from $N(1, 0.01)$, $N(2, 0.01)$, and $N(3, 0.01)$, respectively. Then, $\boldsymbol{m}_i$ is used to generate $\boldsymbol{m}_{ij}$.

Next, the model coefficients for each product are generated based on the model assumption (i.e., underlying models for similar manufacturing processes and products will have similar model structures and coefficients). In the beginning, C orthogonal vectors $\boldsymbol{\eta}_1, \dots, \boldsymbol{\eta}_c, \dots, \boldsymbol{\eta}_C$ are generated based on a normal distribution $N(0, 1)$ as the basis of model coefficients to create C clusters in the simulation. Moreover, the products from the same cluster share the same orthogonal basis of the model coefficients. For example, for Product i, the model coefficient vector $\boldsymbol{\beta}_{0,i}$ is initially generated as: $\boldsymbol{\beta}_{0,i} = \boldsymbol{\eta}_c + \boldsymbol{d}$, where 0 stands for initialization, $\boldsymbol{d}$ is the disturbance vector whose elements are generated from a normal distribution: $d_i \sim N(0,0.1)$. Next, in order to make sure the products in the same cluster have more similar model coefficients, The coefficient matrix is iteratively updated as: $[\boldsymbol{\beta}_{new,1}, \dots, \boldsymbol{\beta}_{new,n}] = [\sum_{j=i}^{n} s_{1j}\boldsymbol{\beta}_{old,j}, \dots, \sum_{j=i}^{n} s_{nj}\boldsymbol{\beta}_{old,j}]$ till convergence is reached, where s_{ij} is the similarity coefficient between Product i and Product j in Eq. (2). Once the Frobenius norm of $[\boldsymbol{\beta}_{i,new}, \dots, \boldsymbol{\beta}_{n,new}] - [\boldsymbol{\beta}_{i,old}, \dots, \boldsymbol{\beta}_{n,old}]$ is smaller than a threshold (say, 0.1 in this research), the iterative update will stop, and the converged $[\boldsymbol{\beta}_{new,1}, \dots, \boldsymbol{\beta}_{new,n}]$ will be used as the underlying model coefficient matrix. In this simulation, the number of features is set to be 951, which is the same as the number of features detailed in Section 3.

Finally, the predictors $\boldsymbol{w}_{i,l}^{\mathrm{T}}$ in Eq. (1) is generated. In order to simulate the signal feature under different manufacturing feature vector, the energy of photodiode signals under different design and setting combinations were collected from physical experiments. Moreover, the manufacturing feature

vector is further used to generate $\boldsymbol{w}_{i,l}^{\mathrm{T}}$ as $\boldsymbol{w}_{i,l}^{\mathrm{T}} = M(\boldsymbol{m}_i) + H(\boldsymbol{m}_i)$, according to $\boldsymbol{m}_i$ in the simulation. $M(\cdot)$ is the mean value of $\boldsymbol{w}_{i,l}^{\mathrm{T}}$ under specific settings, and $H(\cdot)$ is the normal distributed random error term of $\boldsymbol{w}_{i,l}^{\mathrm{T}}$ estimated from the historical data. The RMSEs of simulated data is calculated and compared with the historical data with the same manufacturing feature vector $\boldsymbol{m}_i$. The average normalized RMSE is 0.101. After generating $\boldsymbol{\beta}_{i,l}$ and $\boldsymbol{w}_{i,l}^{\mathrm{T}}$, the underlying model $y_{i,l} = \boldsymbol{w}_{i,l}^{\mathrm{T}}\boldsymbol{\beta}_{i,l} + k \cdot \varepsilon$ is used to generate responses $y_{i,l}$.

For each simulation case, 100 replications of the datasets are generated. The family learning is compared with three benchmark models to evaluate its prediction performance: (1) the Lasso regression (Tibshirani 1996), which should have similar performance with the proposed method when the sample size is large enough; (2) the data-shared Lasso (Gross and Tibshirani 2016), which should have better performance comparing to Lasso regression when the sample size is limited, and (3) the MTL model (Evgeniou and Pontil 2004), which is an effective information shared modeling method.

4.2. Results and discussion

The average prediction RMSEs, R^2 scores, and standard errors (in parenthesis) of simulation studies are shown in Table 2 and Table 3. The values shown in bold are the smallest prediction errors and largest R^2 scores obtained from different models in each simulation case. From the results, the proposed family learning, which considers the similarity between different products, performs the best in quality prediction with both small and large sample size (Case 1, Case 2, Case 3 and Case 4). Even though in simulation Case 2 (No. of the product equals No. of the cluster), where the products are one-of-a-kind,

Table 2. Prediction errors and average standard errors (in parenthesis) over 100 replications.

Cases	Case 1	Case 2	Case 3	Case 4
Lasso w/o m_i	3.56 (0.052)	3.41 (0.050)	3.05 (0.17)	2.98 (0.17)
Lasso w/ m_i	3.56 (0.052)	3.41 (0.050)	3.05 (0.17)	2.98 (0.17)
DSL* w/o m_i	3.24 (0.035)	3.15 (0.038)	2.88 (0.13)	2.91 (0.13)
DSL* w/ m_i	3.24 (0.035)	3.15 (0.038)	2.88 (0.13)	2.91 (0.13)
MTL w/o m_i	3.25 (0.039)	3.06 (0.039)	3.81 (0.18)	3.75 (0.17)
MTL w/ m_i	3.25 (0.039)	3.06 (0.039)	3.81 (0.18)	3.75 (0.17)
FL ** **w/o m_i**	**2.51 (0.022)**	**2.66 (0.022)**	**2.31 (0.11)**	**2.17 (0.11)**
FL ** **w/ m_i**	**2.51 (0.022)**	**2.66 (0.022)**	**2.31 (0.11)**	**2.17 (0.11)**

* DSL is short for data-shared Lasso. ** FL is short for family learning.

Table 3. Average and standard errors (in parenthesis) of R^2 score over 100 replications. The largest scores are highlighted in **bold**.

Cases	Case 1	Case 2	Case 3	Case 4
Lasso w/o m_i	81.3% (0.005)	81.1% (0.004)	83.9% (0.021)	84.2% (0.023)
DSL w/o m_i	83.9% (0.003)	84.4% (0.003)	86.5% (0.013)	86.1% (0.014)
MTL w/o m_i	84.6% (0.003)	84.2% (0.003)	85.9% (0.101)	87.0% (0.101)
FL w/o m_i	**90.7% (0.001)**	**91.3% (0.001)**	**94.7% (0.041)**	**96.9% (0.042)**

the proposed method still yields the best performance compared with the benchmark methods. Because it can reasonably qualify the difference of model coefficients and estimate the similarity measurements among the products.

On the other hand, the MTL and data shared Lasso cannot explain the specific relationship among the products via manufacturing processes and reflect the relationship in model estimation. The Lasso has the worst performance when the sample size is limited (Cases 1 and 2) since it does not consider the similarity among products and manufacturing processes. As the sample size increases (Cases 3 and 4), the performance of the Lasso is competitive with data shared Lasso. In order to identify the prediction

performance with $\boldsymbol{m}_i$ as additional predictors, the authors also added this vector as additional predictors in the model estimation. The improvement of predictions is less than 1% after adding the vector as the predictors. In addition, from the variable selection results, the manufacturing feature vectors are not selected in all cases. Therefore, the advantage of the proposed method comes from the appropriate model structure instead of the benefits of including the information about manufacturing feature vectors.

Table 4**.** Average and standard errors (in parenthesis) of parameter estimation error over 100 replications. The smallest errors are highlighted in **bold**.

Cases	**Case 1**	**Case 2**	**Case 3**	**Case 4**
Lasso	96%	95%	90%	91%
w/o m_i	(0.01)	(0.01)	(0.68)	(0.80)
DSL	96%	93%	40%	43%
w/o m_i	(0.01)	(0.01)	(0.09)	(0.09)
MTL	98%	97%	42%	45%
w/o m_i	(0.02)	(0.02)	(0.10)	(0.10)
FL	**85%**	**79%**	**15%**	**17%**
w/o $\boldsymbol{m_i}$	**(0.008)**	**(0.01)**	**(0.02)**	**(0.04)**

The parameter estimation errors (PE) is defined as: $PE = \frac{\sum_{i=1}^{n}\sum_{k=1}^{p}|\hat{\beta}_{i,k}-\beta_{i,k}|}{\sum_{i=1}^{n}\sum_{k=1}^{p}|\beta_{i,k}|} \times 100\%$ where p is the total number of model parameters; $\hat{\beta}_{i,k}$ is the kth estimated coefficient for Product i; and $\beta_{i,k}$ is the kth true coefficient for Product i. PEs of the four models are summarized in Table 4. Because the manufacturing feature vectors are considered as latent variables, the authors did not compare the PE for the models with $\boldsymbol{m}_i$ (shown as "w/ $\boldsymbol{m}_i$" in Table 2). From the results, it can be found that the proposed family learning also gives the best parameter estimation accuracy in all cases. In the simulation Case 1, the variable selection results are not stable, and the average error of parameter estimation is over 90% (in Table 4). This is mainly because the sample size is much smaller than the number of predictors, and the predictors have strong correlations in Case 1 (and Case 2). It can lead to unstable variable selection results with Lasso regularization terms (Zou and Hastie 2005). For the proposed method, since it considers both the sparsity of the model and similarity of model coefficients among products, it can have a better performance compared with the benchmark methods. Similar to the variable selection results in Case 3 (and Case 4), the sample size is more than Case 1 but still smaller than the number of predictors. There is more information that can be shared among the products in the

model estimation and the variable selection results are much better comparing with the benchmark methods.

Moreover, as shown in Figure 2, the recovered pmf (i.e., similarity measurement) are compared with the ground truth values (due to the space limitation, only simulation Case 4 is presented). It can be observed that even though the clustering pattern among products in Case 4 is the most complicated, the proposed family learning method can still accurately estimate the similarity based on the training dataset.

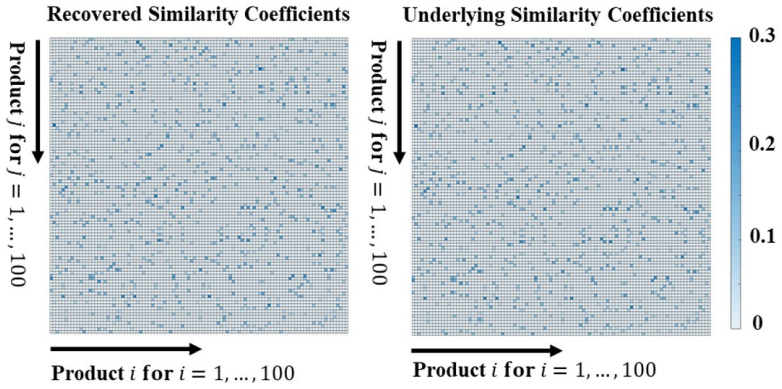

Figure 2. Recovered similarity coefficients (left) and underlying similarity coefficients (right).

5. A Real Case Study

5.1. Experiments setup

In this section, the authors apply the proposed family learning method to a real SLM process in the CAMNet and predict geometric deviations by quantifying the similarity among similar-but-non-identical products and processes. The architecture of the proposed CAMNet testbed is presented in Figure 3 with M connected machines. Assume that *in situ* data of each machine can be collected in real-time. All machines are connected to the data acquisition system via the network. The data acquisition system serves as a middleware to extract the raw data from the machines and communicates with the computation cloud in real-time. Different from the highly intensive algorithms, such as finite element analysis, the *in situ* data-driven prediction model is time-efficient. After the offline training efforts, the online prediction will only take less one second for computation. Therefore, the real-time decisions based on the prediction results can be communicated with the specific machines via MTConnect communication protocol (Vijayaraghavan, et al. 2008). In order to identify the product quality in a CAMNet, modeling multiple similar-but-non-identical AM process is an important step to provide prediction and support the control strategy.

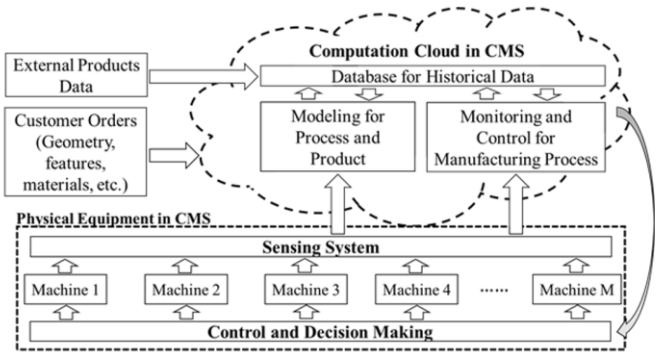

Figure 3. The architecture of the CAMNet testbed.

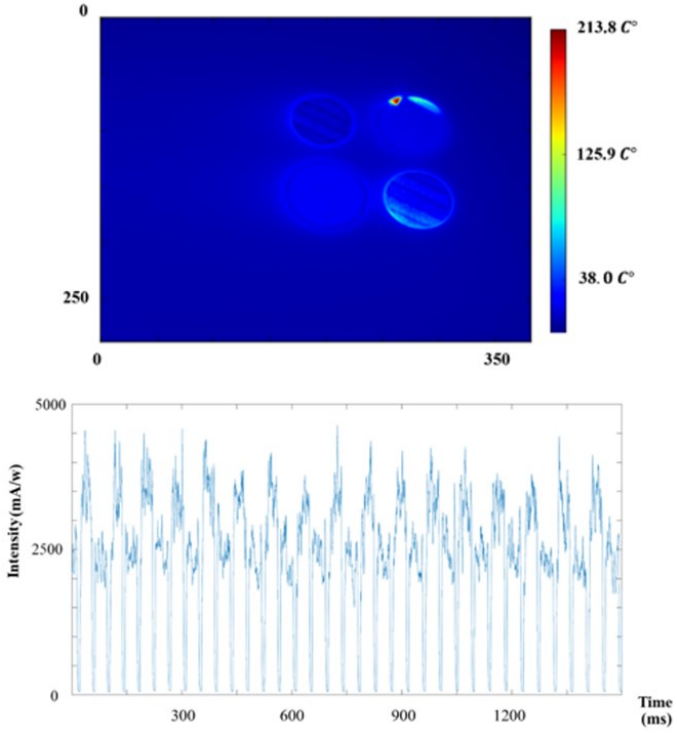

Figure 4. A snapshot of the thermal video (top) and raw photodiode signals (bottom).

The SLM machines in the CAMNet are fully instrumented with sensors and connected to the cloud. Specifically, the data collected from the SLM machine are shown in Figure 4, from Layer $(l-1)$ to Layer l of a SLM process, the sensor system can collect *in situ* photodiode signals and the *in situ* melting video during the melting steps. Both the photodiode system and the thermal camera are mounted on the roof of the chamber. The photodiode system has a wide field of view which covers the entire build area with calibrated distortion. The acquisition frequency of photodiode is 0.1 MHz (signal captured via a NI-GPIO card), that records the laser intensity on the melting pool during the fabrication process. Since the thermal camera is too big to perpendicularly point on the build area, an angle (around

45°) exists between the camera lens and the build area. A camera calibration effort was implemented before we collect the data from the camera (Zhang 2000). The resolution of the thermal image is 382 x 288 pixels with 30Hz framerate. The pixel size of the thermal camera image is around 0.8 millimeters.The sealing performance of the chamber is checked to avoid the oxygen intrusion after the installation of sensors.

The signal features and quality measures are generated to serve as the predictors and responses in the quality-process model. The signals from the photodiode that recorded the laser power are treated as the *in situ* process variables in time-series format. In order to reduce the dimension of raw data, the fast Fourier transformation (FFT) analysis (Welch 1967) is employed to transform the *in situ* signals into the frequency domain. Moreover, the authors chose a frequency band to contain at least 95% of the total energy based on the primary study. Signals in other bands were considered as the noise. As a result, the dimension of predictors was reduced. In total, 951 features are used for model estimation. The scanning path is obtained from the *in situ* thermal videos by calculating the relative x-y positions of the melting pool on the build plate.

After fabricating the products, the layer-to-layer geometries (i.e., contours for each layer) of products are measured via a laser CMM (Coordinate Measuring Machine). A 3D point cloud is generated for each product after obtaining the measurement. The resolution of the CMM machine is 10 microns. As discussed in Methodology, the geometric deviation of the product in each layer is defined as the quality response of the proposed model. As shown in Figure 5, To obtain the geometric deviation for each product in the case study, a 2D point cloud of the product contour in each layer is collected by the laser CMM. It is compared with the corresponding design STL file geometry layer-by-layer according to the layer-wise thickness of the product (Habermann and Kindermann 2007). As shown in Figure 5, in general, the geometric deviation can be defined in the polar coordinates system (Wang, et al. 2018). By mapping the centroids of original CAD design from the STL file and the corresponding product contour in the same polar coordinate, the geometric deviation is defined as the summation of the absolute difference between two radii r_i and r_i' on specific angle α_i.

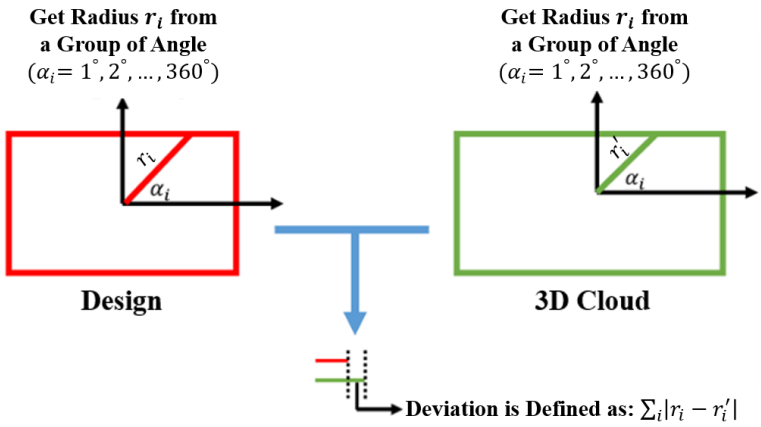

Figure 5. The way to extract geometric deviations as quality measures.

In the proposed CAMNet, the product design, laser power, scanning speed, and hatch distance are selected as factors in the design of experiments. As shown in Figure 6, two different product designs (i.e., square design and cylinder design) are investigated in experiments due to a limited budget, such that sufficient samples for each cluster can be obtained. A fractional factorial design (Montgomery 2017) with three levels of settings and two kinds of designs is conducted to test the modeling performance. The levels of setting variables in the experiment are shown in Table 5. In total, there are 18 different combinations of setting variables and designs with two replications in the experiment. Limited by the size of the powder bed, these 36 products are manufactured in three builds. The products were fabricated using Inconel 718 powder in an EOS M290 commercial SLM machine. For the square design, the side length is 2 centimeters, the radius of the curvature is 0.5 centimeters. For the cylinder design, the radius of the cylinder is 1 centimeter, the radius of the cone is 0.75 centimeters, and the slant height of the cone is 1.5 centimeters.. The products were randomly assigned to pre-defined locations on the powder bed.

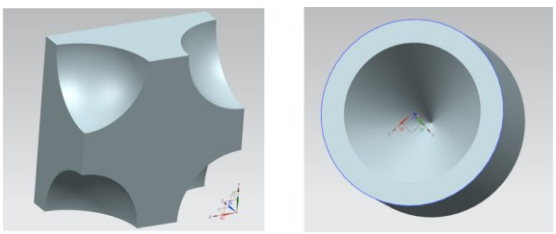

Figure 6. Examples for a square design (left) and a cylinder design (right).

Table 5. The experiment design parameters.

Parameters	Units	Level 1	Level 2	Level 3
Laser Power	W	170	195	220
Scan Speed	mm/s	983	1083	1183
Hatch Distance	mm	0.07	0.09	0.11

To better demonstrate the implementation procedures for the proposed method in a CAMNet, we provide a flowchart in Figure 7. Before deploying the fabrication task into specific systems in the CAMNet, the design features and setting variables for the process will be collected, and the initial manufacturing feature vector will be obtained for each product. Moreover, the scanning path and three setting variables are used to generate the manufacturing feature vector for each product in the experiment. Before deploying the fabrication task into specific systems in the CAMNet, the design features and setting variables for all process (i.e., both historical and new processes) will be collected to generate manufacturing feature vector (remark: it consists of the process setting variables and summary statistics of the scanning path for each product). After standardization to ensure $\sum_j \frac{1}{s_{ij}} = 1, (i \neq j)$, the Euclidean distance (Danielsson 1980) between two manufacturing feature vectors $(\boldsymbol{m}_i, \boldsymbol{m}_j)$ is denoted as the initial similarity coefficient: $s'_{ij} = E(\boldsymbol{m}_i, \boldsymbol{m}_j)$. The model coefficients of the historical product, which has the most similar manufacturing features (i.e., smallest Euclidean distance) as the new product, will be treated as the initial model coefficients for the new product to enhance quality prediction at the beginning of the process. During the fabrication, the family learning model will be estimated, and the layer-to-layer quality prediction will be implemented. The prediction results can provide the necessary information for the online layer-to-layer control (Wang, Jin and Henkel 2018). Furthermore, the model will be iteratively updated based on the new observations from the CAMNet. After the fabrication, the quality variables, *in situ* variables, model coefficients, and manufacturing feature vectors will be recorded and used to estimate the family learning model for a brand-new personalized product.

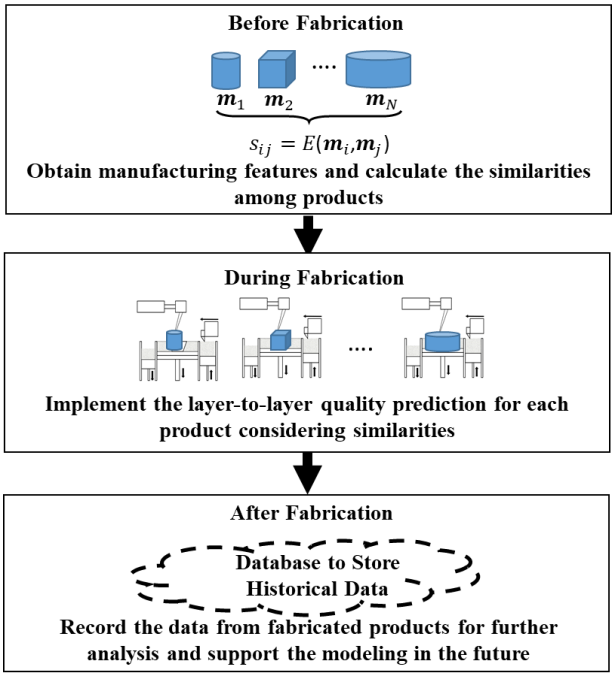

Figure 7. Family learning for layer-to-layer quality prediction in a CAMNet.

5.2. Results and discussion

The RMSE is used to evaluate the accuracy of the model since it generally represents the magnitude of error between the predicted overall product geometric deviation and the real deviation which is measured by the laser CMM. The family learning model is compared with three benchmark models as described in the simulation study. There are two testing scenarios: (1) layer-to-layer modeling and (2) leave-one-product-out. In the layer-to-layer modeling scenario, data from all previous layers are used to estimate the family learning model, and the data from the next layer are used to test the accuracy of the prediction. Note that the scanning path and process settings are known before the modeling, which will be used to generate the manufacturing feature vectors. To improve the model estimation, the model coefficients are re-estimated per every six layers. For this layer-to-layer modeling scenario, there are 54 layers obtained from the case study. On the other hand, in the leave-one-product-out scenario, eleven products in one build are used as the training dataset, and the remaining one product is treated as the testing dataset in sequence. In particular, for the proposed method, data shared lasso and MTL, the model coefficient for the testing product is selected from the training product which has the most similar design and setting (i.e., the Euclidean distance between their manufacturing feature vectors) with the testing product. For Lasso regression, since it has the same model coefficient for every product, we do

not need to specify the model coefficient we use for the testing product. The leave-one-product-out scenario can be considered as a brand-new product from the CAMNet. We want to check whether the historical data from other products can provide an accepted model to implement quality prediction without the sample from the new product.

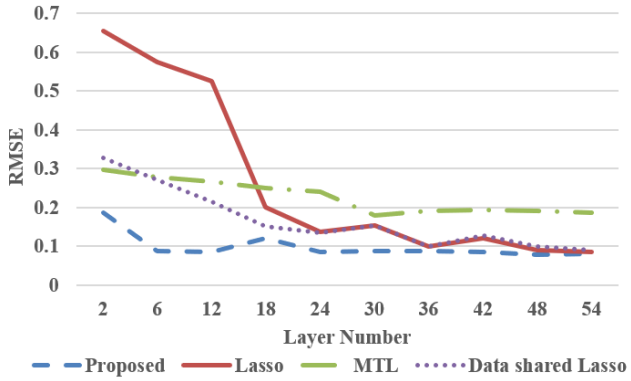

Figure 8. Testing RMSEs of the proposed model and benchmark models.

Table 6. R^2 score for the proposed model and benchmark models.

Model Name	FL	Lasso	DSL	MTL
R^2 score (12 Layers)	90%	73%	81%	84%
R^2 score (44 Layers)	92%	93%	92%	86%

RMSEs of the testing dataset in layer-to-layer modeling are summarized in Figure 8. The R^2 scores for two scenarios (i.e., sample size is limited (12 Layers), and sample size is adequate (44 Layers) are also shown in Table 6. The results of the layer-to-layer modeling scenario indicate that the proposed method has better prediction performance compared with three benchmarks. It can be observed that the testing RMSEs have a decreasing trend for all four models when the number of layers is increased. Lasso regression has the worst prediction performance when the sample size is limited since it requires relatively larger sample sizes and does not share information with other products. The data shared Lasso has better performance but worse than the proposed method when the sample size is limited. In addition, the data shared Lasso presents the competitive results with Lasso when the number of printed layers is becoming larger. The prediction errors of MTL are consistently larger than the proposed method since it does not consider similarity among the products. By estimating the similarity measurements among products and further using the similarity measurement to penalize their model

coefficients, the proposed method yields the best prediction performance when the sample size is limited and remains the best model with the lowest prediction errors. The results showed the capabilities of the proposed method to model similar-but-non-identical connected systems.

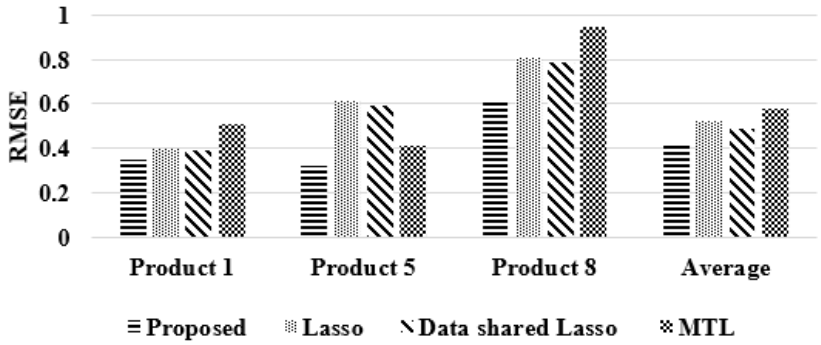

Figure 9. Leave-one-product-out testing RMSEs.

The results of the leave-one-product-out scenario are shown in Figure 9. The proposed model results in the lowest testing RMSEs on average compared with the benchmark results. The data shared Lasso outperforms the Lasso since it partially considers the information from other products. Because the leave-one-product-out scenario includes the data from 54 layers, the Lasso regression has adequate samples to estimate an acceptable model. Therefore, the average error of Lasso is smaller than MTL. For each product, MTL and data shared Lasso consistently have greater testing errors than the proposed model since the proposed model forces similar products to have similar model coefficients. These leave-one-product-out results show the potential of the proposed method in modeling personalized products. Even though the data for a specific product is unavailable, the CAMNet still can use the model coefficients from the historical product which has the most similar design and setting with the new product to implement modeling efforts with acceptable performance. Therefore, based on the manufacturing feature vector extracted from the new product and the corresponding similarity coefficient (i.e., $s'_{ij} = E(\boldsymbol{m}_i, \boldsymbol{m}_j)$), family learning method can quantitatively identify the amount of information that can be borrowed from other products and improve the model accuracy for the new product when the sample size is limited. In general, there is no specific boundary condition for the similarity in the proposed model. However, there are also existed limitations of the proposed model. If a product has a very different product and process design compared with other products, the proposed method might not be efficient to accurately predict the quality variable with limited sample size for this

product. It is because in such a case, very limited information (sample size) is available to support the modeling for this product, and there is also limited information that can be shared from other product due to the differences in product and process design. It is worth mentioning that the family learning method will result in a similar quality prediction accuracy compared with the Lasso regression, which only utilizes the information from the individual product.

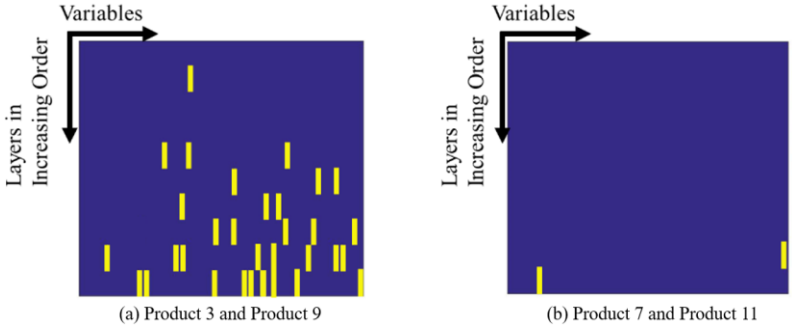

Figure 10. Variable selection results from the layer-to-layer modeling scenario.

Assumptions of the proposed model are validated in variable selection results. To compare the variable selection results among products in the layer-to-layer modeling scenario, differences of variable selection between product 3 (cylinder design with laser power 170W, scan speed 983mm/s and hatch distance 0.07mm) and product 9 (square design with laser power 220W, scan speed 1183mm/s and hatch distance 0.11mm), which are shared the least similarity; and between product 7 (square design with laser power 195W, scan speed 1083mm/s and hatch distance 0.07mm) and product 11 (square design with laser power 195W, scan speed 1083mm/s and hatch distance 0.09mm), which are shared the most similarity are shown in Figure 10. Each row represents the importance of a specific variable, and each column represents the model sequence in the layer-to-layer scenario. The lighter (yellow) color represents the difference between the significant variables of model coefficients for two products. As shown in Figure 9(a), when the two products and corresponding manufacturing process settings are different, their models have different significant variables. On the other hand, the significant variables of the models for two similar products are consistent as shown in Figure 9 (b). This variable selection result validates the Assumptions (1) and (2) in Section 3, i.e., the process similarity can be partially reflected by the designs and process settings, which reflect similarity on the underlying model structure and coefficients. Therefore, when modeling a new product in a SLM process, the samples from other

similar products can be also introduced to potentially improve the accuracy of the model for the new product. This approach can significantly reduce the sample size requirements for modeling a brand-new product and process designs in a CAMNet. Furthermore, Furthermore, the Assumptions (3), i.e., the linear model assumption is also checked via residual plots. For example, the residual plots of product 3 are shown in Figure 11. Residuals from other models are also validated for the linear model assumption. From Figure 11, it can be concluded that the residuals roughly follow the normal distribution; the average of the residuals is close to zero; the residuals have a constant variance; and the residuals are roughly independent. In summary, the prediction for the family learning with laser related in *in situ* variables is adequate as the can explain the highest proportion of the variance of geometric deviations with limited information left in unexplained variance.

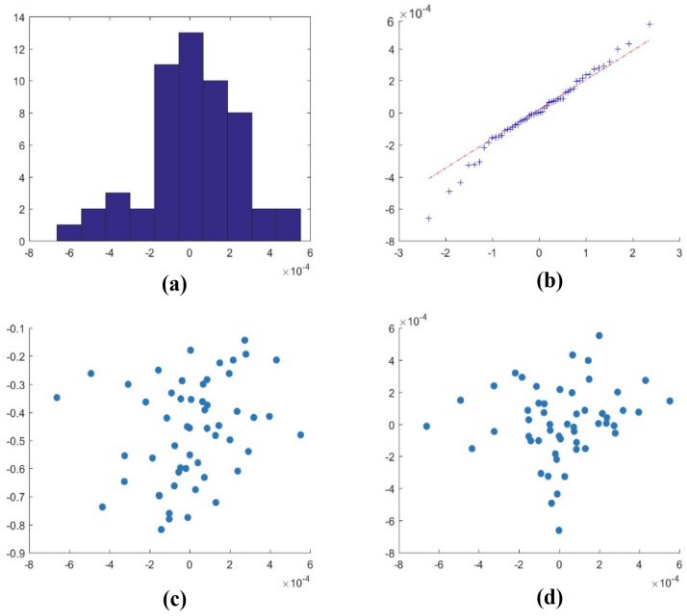

Figure 11. Linear Model Assumption Check for the Proposed Model ((a) Histogram of Residual; (b) Q-Q plot; (c) Residual vs. Fitting Value; (d) $\hat{\varepsilon}_t$ vs. $\hat{\varepsilon}_{t-1}$).

6. Conclusion

The CAMNet connects AM facilities as a network with computation resources integrated to provide responsive computation services for manufacturing process modeling, diagnosis, prognosis, and control. However, the increasing demands of personalized design and unique processes can significantly reduce the performances of most traditional data-driven modeling methods due to the limited sample size. Therefore, we proposed a family learning method to quantify the amount of shared

information by jointly learning the multitask model coefficients for each product and the similarity measurements. The proposed method was validated in a simulation and a case study in SLM processes. The results showed that the proposed family learning model outperforms Lasso regression (Tibshirani 1996), data shared Lasso (Gross and Tibshirani 2016), and MTL (Evgeniou and Pontil 2004), especially when the sample size is limited. Further analysis of variable selection results demonstrated the effectiveness of the proposed information sharing method and reveals the underlying manufacturing similarity via the variable selection results. The proposed family learning showed the potential to apply to other industrial applications for system quality prediction (Sun, et al. 2016), prognosis health management (Song and Liu 2018, Fang, et al. 2017), and etc.

This paper leads to several future research questions. First, the experiments in this paper focus on one SLM machine with combinations of product designs and process settings. We will extend the proposed method to similar-but-non-identical SLM machines, with different types of materials, multiple *in situ* variables (e.g., gas flow speed, recoating speed) and performance variables (e.g., porosity, mechanical properties) in a CAMNet. Moreover, the modeling for local region quality will also investigate. Second, by considering the benefit from the online adjustable capability for the AM process that can control the process during the fabrication via the hardware protocol such as MTconnect (Vijayaraghavan, et al. 2008), an extension of the family learning method will be investigated to process monitoring (Jin and Liu 2013), diagnosis (Seshadrinath, Singh and Panigrahi 2013), and control (Huang, Liu, Chalivendra, Ceglarek and Kong 2014) in a CMS. Finally, the proposed linear model structure can be improved to quantify more complex variable relationships. Examples can be found in a functional graphical model (Sun, et al. 2017), a nonlinear model (Shumway and Stoffer 2011), or quantitative and qualitative model (Deng and Jin 2015).

Acknowledgment

This work was supported in part by the National Science Foundation (Grant No. CMMI-1634867).

References

Lee, J., Bagheri, B. and Jin, C. (2016) Introduction to cyber manufacturing. Manufacturing Letters,**8**, 11-15.

Yang, H., Kumara, S., Bukkapatnam, S.T. and Tsung, F. (2019) The internet of things for smart manufacturing: A review. IISE Transactions, 1-27.

Hu, S.J. (2013) Evolving paradigms of manufacturing: from mass production to mass customization and personalization. Procedia Cirp,**7**, 3-8.

Chen, X., Wang, L., Wang, C. and Jin, R. (2018) Predictive offloading in mobile-fog-cloud enabled cyber-manufacturing systems, in Proceedings of 2018 IEEE Industrial Cyber-Physical Systems (ICPS), 167-172.

Zhang, J., Song, B., Wei, Q., Bourell, D. and Shi, Y. (2019) A review of selective laser melting of aluminum alloys: Processing, microstructure, property and developing trends. Journal of Materials Science & Technology,**35**, 270-284.

Seabra, M., Azevedo, J., Araújo, A., Reis, L., Pinto, E., Alves, N., Santos, R. and Mortágua, J.P. (2016) Selective laser melting (SLM) and topology optimization for lighter aerospace componentes. Procedia Structural Integrity,**1**, 289-296.

Wei, Q., Li, S., Han, C., Li, W., Cheng, L., Hao, L. and Shi, Y. (2015) Selective laser melting of stainless-steel/nano-hydroxyapatite composites for medical applications: Microstructure, element distribution, crack and mechanical properties. Journal of Materials Processing Technology, **222**, 444-453.

Gibson, I., Rosen, D. and Stucker, B. (2010) Additive manufacturing technologies, Springer US.

Frazier, W.E. (2014) Metal additive manufacturing: A review. Journal of Materials Engineering and Performance,**23**, 1917-1928.

Strano, G., Hao, L., Everson, R.M. and Evans, K.E. (2013) Surface roughness analysis, modelling and prediction in selective laser melting. Journal of Materials Processing Technology, **213**, 589-597.

Spears, T.G. and Gold, S.A. (2016) In-process sensing in selective laser melting (SLM) additive manufacturing. Integrating Materials and Manufacturing Innovation, **5**, 16-40.

Chen, X., Wang, L., Wang, C., and Jin, R. (2018) Predictive offloading in mobile-fog-cloud enabled cyber manufacturing systems, in Proceedings of the 1st IEEE International Conference on Industrial Cyber-Physical Systems.

Olakanmi, E.O., Cochrane, R.F. and Dalgarno, K.W. (2015) A review on selective laser sintering/melting (SLS/SLM) of aluminium alloy powders: Processing, microstructure, and properties. Progress in Materials Science,**74**, 401-477.

Evgeniou, T. and Pontil, M. (2004) Regularized multi--task learning, in Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 109-117.

Chen, X., Sun, H. and Jin, R. (2016) Variation analysis and visualization of manufacturing processes via augmented reality, in Proceedings of Industrial and Systems Engineering Research Conference 2016.

Wang, L., Jin, R. and Henkel, D. (2018) Data fusion for in situ layer-wise modeling and feedforward control of selective laser melting processes, in Proceedings of IISE Annual Conference 2018.

Tibshirani, R.J. (1996) Regression shrinkage and selection via the LASSO. J R Stat Soc B. Journal of the Royal Statistical Society,**58**, 267-288.

Gross, S.M. and Tibshirani, R. (2016) Data shared lasso: a novel tool to discover uplift. Computational Statistics & Data Analysis,**101**, 226-235.

Wang, D., Liu, J. and Srinivasan, R. (2010) Data-driven soft sensor approach for quality prediction in a refining process. IEEE Transactions on Industrial Informatics,**6**, 11-17.

Jin, R. and Liu, K. (2013) Multimode variation modeling and process monitoring for serial-parallel multistage manufacturing processes. IIE Transactions,**45**, 617-629.

Seshadrinath, J., Singh, B. and Panigrahi, B.K. (2013) Vibration analysis based interturn fault diagnosis in induction machines. IEEE Transactions on Industrial Informatics,**10**, 340-350.

Huang, W., Liu, J., Chalivendra, V., Ceglarek, D. and Kong, Z. (2014) Statistical modal analysis for variation characterization and application in manufacturing quality control. IIE Transactions,**46**, 497-511.

Zwolenski, M. and Weatherill, L. (2014) The digital universe rich data and the increasing value of the internet of things. Australian Journal of Telecommunications & the Digital Economy,2.

Li, P. and Chen, S. (2017) Hierarchical gaussian processes model for multi-task learning. pattern recognition,74.

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

Rao, P., Liu, J., Roberson, D., Kong, Z. and Williams, C. (2015) Online real-time quality monitoring in additive manufacturing processes using heterogeneous sensors. Journal of Manufacturing Science & Engineering,**137**, 1007-1 - 1007-12.

Khanzadeh, M., Tian, W., Yadollahi, A., Doude, H.R., Tschopp, M.A. and Bian, L.J.A.M. (2018) Dual process monitoring of metal-based additive manufacturing using tensor decomposition of thermal image streams.**23**, 443-456.

Icten, E., Nagy, Z.K., Reklaitis, G.V.J.C. and Engineering, C. (2015) Process control of a dropwise additive manufacturing system for pharmaceuticals using polynomial chaos expansion based surrogate model.**83**, 221-231.

Xing, F., Liu, W. and Wang, T. (2006) Real-time sensing and control of metal powder laser forming, in Proceedings of Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on, IEEE, pp. 6661-6665.

Krauss, H., Zeugner, T. and Zaeh, M.F. (2015) Thermographic process monitoring in powderbed based additive manufacturing, in Proceedings of AIP Conference Proceedings, AIP, pp. 177-183.

Sun, H., Rao, P.K., Kong, Z., Deng, X. and Jin, R. (2017) Functional quantitative and qualitative models for quality modeling in a fused deposition modeling process. IEEE Transactions on Automation Science & Engineering, pp. 1-11.

Shevchik, S., Kenel, C., Leinenbach, C. and Wasmer, K. (2018) Acoustic emission for in situ quality monitoring in additive manufacturing using spectral convolutional neural networks. Additive Manufacturing,**21**, 598-604.

Yadroitsev, I., Krakhmalev, P. and Yadroitsava, I. (2014) Selective laser melting of Ti6Al4V alloy for biomedical applications: Temperature monitoring and microstructural evolution. Journal of Alloys and Compounds,**583**, 404-409.

Huang, Q., Wang, Y., Lyu, M. and Lin, W. (2020) Shape deviation generator--a convolution framework for learning and predicting 3-d printing shape accuracy. IEEE Transactions on Automation Science and Engineering.

Jin, Y., Qin, S.J. and Huang, Q. (2020) Modeling inter-layer interactions for out-of-plane shape deviation reduction in additive manufacturing. IISE Transactions,**52**, 721-731.

Ferreira, R.d.S.B., Sabbaghi, A. and Huang, Q. (2019) Automated geometric shape deviation modeling for additive manufacturing systems via Bayesian neural networks. IEEE Transactions on Automation Science and Engineering,**17**, 584-598.

Sabbaghi, A. and Huang, Q. (2018) Model transfer across additive manufacturing processes via mean effect equivalence of lurking variables. The Annals of Applied Statistics,**12**, 2409-2429.

Francis, J., Sabbaghi, A., Ravi Shankar, M., Ghasri-Khouzani, M. and Bian, L. (2020) Efficient distortion prediction of additively manufactured parts using bayesian model transfer between material systems. Journal of Manufacturing Science and Engineering,142.

Sabbaghi, A., Huang, Q. and Dasgupta, T. (2018) Bayesian model building from small samples of disparate data for capturing in-plane deviation in additive manufacturing. Technometrics,**60**, 532-544.

Cheng, L., Tsung, F. and Wang, A. (2017) A statistical transfer learning perspective for modeling shape deviations in additive manufacturing. IEEE Robotics and Automation Letters,**2**, 1988-1993.

Cheng, L., Wang, K. and Tsung, F. (2020) A hybrid transfer learning framework for in-plane freeform shape accuracy control in additive manufacturing. IISE Transactions, 1-15.

Kontar, R., Raskutti, G. and Zhou, S. (2020) Minimizing negative transfer of knowledge in multivariate gaussian processes: a scalable and regularized approach. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Bhandari, S. and Lopez-Anido, R. (2018) Finite element analysis of thermoplastic polymer extrusion 3D printed material for mechanical property prediction. Additive Manufacturing,**22**, 187-196.

Chen, Q., Liang, X., Hayduke, D., Liu, J., Cheng, L., Oskin, J., Whitmore, R. and To, A.C. (2019) An inherent strain based multiscale modeling framework for simulating part-scale residual deformation for direct metal laser sintering. Additive Manufacturing,**28**, 406-418.

Olleak, A. and Xi, Z. (2020) Calibration and validation framework for selective laser melting process based on multi-fidelity models and limited experiment data. Journal of Mechanical Design,142.

Li, J., Jin, R. and Hang, Z.Y. (2018) Integration of physically-based and data-driven approaches for thermal field prediction in additive manufacturing. Materials & Design,**139**, 473-485.

Wang, L., Chen, X., Kang, S., Deng, X. and Jin, R. (2020) Meta-modeling of high-fidelity FEA simulation for efficient product and process design in additive manufacturing. Additive Manufacturing, 101211.

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

Bishop, C.M. (2006) Pattern recognition and machine learning, springer.

De Leeuw, J. (1994) Block-relaxation algorithms in statistics, in Information systems and data analysis, Springer, 308-324.

Lange, K. (2010) Numerical analysis for statisticians, Springer Science & Business Media.

Forsgren, A., Gill, P.E. and Wright, M.H. (2002) Interior methods for nonlinear optimization. SIAM review,**44**, 525-597.

Mehrotra, S. (1992) On the implementation of a primal-dual interior point method. SIAM Journal on optimization,**2**, 575-601.

Zhou, J., Chen, J. and Ye, J. (2011) Malsar: Multi-task learning via structural regularization. Arizona State University,21.

Chen, X., Pan, W., Kwok, J.T. and Carbonell, J.G. (2009) Accelerated gradient method for multi-task sparse learning problem, in Proceedings of 2009 Ninth IEEE International Conference on Data Mining, IEEE, pp. 746-751.

Nesterov, Y. (2008) Accelerating the cubic regularization of Newton's method on convex problems. Mathematical Programming,**112**, 159-181.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005) Sparsity and smoothness via the fused lasso. Journal of the Royal Statistical Society: Series B (Statistical Methodology),**67**, 91-108.

Montgomery, D.C. (2017) Design and analysis of experiments, John wiley & sons.

Zou, H. and Hastie, T. (2005) Zou H, Hastie T. Regularization and variable selection via the elastic net. J R Statist Soc B. 2005;67(2):301-20. Journal of the Royal Statistical Society,**67**, 301-320.

Vijayaraghavan, A., Sobel, W., Fox, A., Dornfeld, D. and Warndorf, P. (2008) Improving machine tool interoperability using standardized interface protocols: MT Connect. Laboratory for Manufacturing & Sustainability.

Zhang, Z. (2000) A flexible new technique for camera calibration. IEEE Transactions on pattern analysis and machine intelligence,**22**, 1330-1334.

Welch, P.D. (1967) The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. IEEE Trans. Audio & Electroacoust., Volume AU-15, p. 70-73,**15**, 70-73.

Habermann, C. and Kindermann, F. (2007) Multidimensional spline interpolation: Theory and applications. Computational Economics,**30**, 153-169.

Wang, L., Jin, R. and Henkel, D. (2018) Data fusion for in situ layer-wise modeling and feedforward control of selective laser melting processes, in Proceedings of the IISE Annual Conference City.

Danielsson, P.E. (1980) Euclidean distance mapping. Computer Graphics & Image Processing,**14**, 227-248.

Sun, H., Deng, X., Wang, K. and Jin, R. (2016) Logistic regression for crystal growth process modeling through hierarchical nonnegative garrote-based variable selection. IIE Transactions,**48**, 787-796.

Song, C. and Liu, K. (2018) Statistical degradation modeling and prognostics of multiple sensor signals via data fusion: A composite health index approach. IISE Transactions,**50**, 853-867.

Fang, X., Gebraeel, N.Z. and Paynabar, K. (2017) Scalable prognostic models for large-scale condition monitoring applications. IISE Transactions,**49**, 698-710.

Vijayaraghavan, A., Sobel, W., Fox, A., Dornfeld, D. and Warndorf, P. (2008) Improving machine tool interoperability using standardized interface protocols: MT connect.

Sun, H., Huang, S. and Jin, R. (2017) Functional graphical models for manufacturing process modeling. IEEE Transactions on Automation Science & Engineering,**14**, 1612-1621.

Shumway, R.H. and Stoffer, D.S. (2011) State-Space Models, Springer New York.

Deng, X. and Jin, R. (2015) QQ Models: Joint modeling for quantitative and qualitative quality responses in manufacturing systems. Technometrics,**57**, 320-331.