IISE Transactions

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uiie21

Transformer-enabled generative adversarial imputation network with selective generation (SGT-GAIN) for missing region imputation

Yuxuan Li, Zhangyue Shi & Chenang Liu

To cite this article: Yuxuan Li, Zhangyue Shi & Chenang Liu (02 May 2023): Transformer-enabled generative adversarial imputation network with selective generation (SGT-GAIN) for missing region imputation, IISE Transactions, DOI: 10.1080/24725854.2023.2193257

To link to this article: https://doi.org/10.1080/24725854.2023.2193257

IISE TRANSACTIONS
https://doi.org/10.1080/24725854.2023.2193257

Transformer-enabled generative adversarial imputation network with selective generation (SGT-GAIN) for missing region imputation

Yuxuan Li, Zhangyue Shi, and Chenang Liu

The School of Industrial Engineering & Management, Oklahoma State University, Stillwater, OK, USA

ABSTRACT

Although data have been extensively leveraged for process monitoring and control in advanced manufacturing, it still suffers from the connection issues among sensors, machines, and computers, which may lead to significant data loss, i.e., missing region in the collected data, in the application of data-driven monitoring. To address the missing region issues, one popular way is to perform missing data imputation. With the advances of machine learning, many approaches have been developed for the missing data imputation, such as the popular Generative Adversarial Imputation Network (GAIN), which is based on the Generative Adversarial Network (GAN). However, the inherent shortcomings of generative adversarial architecture may still lead to unstable training. More importantly, the collected online sensor data in manufacturing are in sequential order whereas GAIN considered the input data independently. Hence, to address these two limitations, this work proposes a novel approach termed transformer-enabled GAIN with selective generation (SGT-GAIN). The contributions of the proposed SGT-GAIN consist of three aspects: (i) the architecture for transformer-enabled generation is developed to capture the sequential information among the data; (ii) a selective multi-generation framework is proposed to further reduce the imputation bias; and (iii) an ensemble learning framework is applied to enhance the imputation robustness. Both the numerical simulation study and a real-world case study in additive manufacturing demonstrated the effectiveness of the proposed SGT-GAIN.

ARTICLE HISTORY
Received 31 July 2022
Accepted 7 March 2023

KEYWORDS
Advanced manufacturing; generative adversarial imputation network (GAIN); missing region imputation; selective generation; transformer

1. Introduction

With the recent advancements in sensor technologies, more and more manufacturing systems have become data-enabled, which makes great contributions to the improvement of monitoring and control. For instance, in additive manufacturing, heterogeneous sensors can be mounted and the collected sensor signals can be leveraged to identify the real-time process conditions (Rao *et al.*, 2015; Lu and Wong, 2018; Liu *et al.*, 2020; Liu *et al.*, 2021). Specifically, when unexpected process errors or anomalies occur, the monitoring models can detect the patterns from the collected data and then provide alarms. Although modern information technologies have significantly improved the efficiency and trustworthiness of data transmission, it is still possible that the connection between sensors and machines is not good enough, such as the common poor connection or even miss connection, leading to data loss for analysis, i.e., the missing region issue. Under such circumstances, to accurately impute the missing regions would be greatly beneficial for data analysis. As described in Figure 1, the missing region may lead to significant bias to monitor the processes, since it is not possible to send the complete data to the trained monitoring model. Hence, it is critically needed to address the missing region issue in the collected data, and thereby, facilitate the monitoring performance in manufacturing systems. As the missing region issue can be treated as a specific type of incomplete data issue, one of the most popular directions to address the data incompleteness issue is to perform the appropriate data imputation techniques (Lakshminarayan *et al.*, 1996; Vangipuram *et al.*, 2020).

In recent decades, many advanced imputation approaches have been developed for missing data imputation. Specifically, the imputation approaches can be classified into two groups: the conventional approaches and machine learning-based approaches (Mirzaei *et al.*, 2022). For the conventional approaches, the statistics-based imputation (Musil *et al.*, 2002), matrix completion (Mazumder *et al.*, 2010), and the expectation maximization algorithm (García-Laencina *et al.*, 2010) are widely applied. However, although they are easy to calculate, the sequential information in the data may not be considered, which does not fit with the sequentially collected sensor signals. In addition, for some of the machine learning-based methods, such as the k nearest neighbors (k-NN) (Zhang, 2012), MissForest (Stekhoven and Bühlmann, 2012), they may also have the similar limitations. Hence, the neural network-based imputation approaches, such as Denoising AutoEncoder (DAE) (Vincent *et al.*, 2008), and Generative Adversarial Nets (GAN) (Goodfellow *et al.*, 2020)-based imputation methods, become popular. With the neural network structures, the collected data can be transformed as window-based samples, so that the sequential information in

CONTACT Chenang Liu chenang.liu@okstate.edu
Supplemental data for this article is available online at https://doi.org/10.1080/24725854.2023.2193257

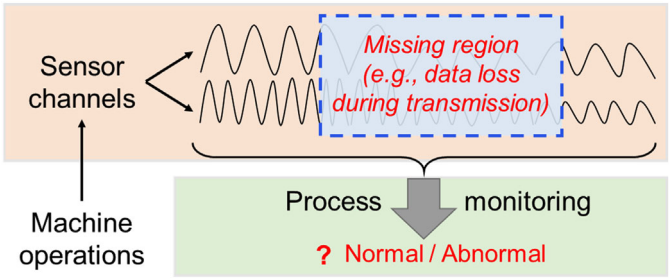

Figure 1. A demonstration of the missing region in the collected manufacturing data.

the window could be efficiently utilized (Li *et al.*, 2021). However, for the DAE, a complete dataset is required to train the model, but it may be very challenging to obtain a complete dataset in practice. Therefore, most of the existing imputation approaches are not suitable for this study. Under such circumstances, GAN-based approaches stand out, since it is able to learn the distribution of the data instead of only performing the imputation. Kim *et al.* (2020) has provided a detailed survey about GAN-based imputation approaches, which includes the Generative Adversarial Imputation Nets (GAIN) (Yoon *et al.*, 2018), Stackelberg GAN (Zhang and Woodruff, 2018), and Collaborative GAN (Lee *et al.*, 2019). Particularly, GAIN (Yoon *et al.*, 2018; Dogan *et al.*, 2023) is widely applied, due to its superior performance.

However, GAIN also has some critical shortcomings to accomplish missing region imputation. Particularly, although the real-time sensor data can be sent to GAIN, how to learn the underlying complex sequential information still remains challenging. In addition, in the imputation process of GAIN, the generated values for a non-missing area may be significantly different from actual values, which may lead to imputation bias. In addition, due to the inherent properties of GAN architecture, the training process of GAIN may also be unstable. Therefore, to bridge the above-mentioned gaps, a new imputation approach termed transformer-enabled GAIN with selective generation (SGT-GAIN) is proposed, and its main contributions consist of: (i) a transformer-enabled architecture is incorporated to capture the sequential information in the window-based samples; (ii) a selective multi-generation framework is proposed to select high-quality imputations and reduce the imputation bias; and (iii) an ensemble learning framework is applied to further enhance the robustness of the imputation model.

The rest of this article is structured as follows. The missing region issue is defined and the GAIN is introduced in Section 2. Then the proposed research methodology is discussed in Section 3. Afterwards, the simulation study and a real-world Additive Manufacturing (AM) case study are conducted in Secion 4. Finally, Section 5 summarizes the conclusions of this study.

2. Problem statement and research background

2.1 Missing region issue

Assume that the sensors involve d channels and s samples are collected. Then the multivariate time series can be represented as a data matrix $\mathbf{X}_t$ following $\mathbb{R}^{s\times d}$. As described in Figure 2, through time window-based sampling with window size n ($n = 3$ in Figure 2), $\mathbf{X}_t$ is transformed to data tensor $\mathcal{X}_W$ following $\mathbb{R}^{p\times n\times d}$ where p is the number of samples after window-based sampling.

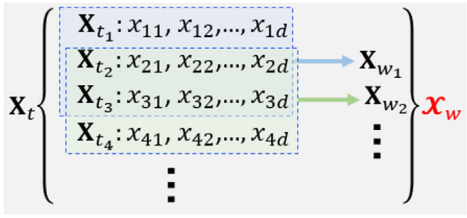

Figure 2. A demonstration of window-based sampling for online sensor data.

Define the mask matrix $\mathcal{M}_W$ in $\mathbb{R}^{p\times n\times d}$ as a binary matrix to demonstrate the missing region. Assume that the size of one missing region is $r \times d$ in each window and the first missing sample in the ith window is f_i-th sample. Then the elements in $\mathcal{M}_W$ are shown as (1). In this way, by using $\odot$ for element-wise multiplication, $\mathcal{M}_W \odot \mathcal{X}_W$ could represent the existing values in $\mathcal{X}_W$ whereas $(1 - \mathcal{M}_W) \odot \mathcal{X}_W$ could represent the missing regions in $\mathcal{X}_W$. The problem is to impute the $(1 - \mathcal{M}_W) \odot \mathcal{X}_W$, and the time-dependent information within each window-based sample could be considered for the imputation:

$$\mathcal{M}_{Wijk} = \begin{cases} 0 & \text{If } f_i \le j \le f_i + r \\ 1 & \text{Otherwise} \end{cases} \quad i = 1, \ldots, p; j = 1, \ldots, n; k = 1, \ldots, d \tag{1}$$

2.2 GAIN

As discussed in Section1, this proposed method is driven by the GAIN, since GAIN has demonstrated its superior performance in missing data imputation (Yoon *et al.*, 2018). Following the popular GAN architecture (Goodfellow *et al.*, 2020), GAIN also involves two key components, the generator G and the discriminator D. G generates the fake data whereas D makes decisions to consider whether the input data are generated data or actual data. In short, G and D compete with each other. However, instead of sending the noise and actual data for training in GAN, three different matrices are used in GAIN, i.e., data matrix $\mathbf{X}$, mask matrix $\mathbf{M}$, and hint matrix $\mathbf{H}$. As described in Section 2.1, $\mathbf{X}$ records the actual values of one window. $\mathbf{M}$ describes the location of missing regions. As for $\mathbf{H}$, it passes some hint information to the discriminator D, which is controlled by the hint parameter h.

The output of G is denoted as $\overline{\mathbf{X}}$ with noise $\mathbf{Z}$. It is also a matrix following $\mathbb{R}^{n\times c}$, similar to $\mathbf{X}$ based on inputting noise $\mathbf{Z}$. In each iteration, as shown in (2), $\overline{\mathbf{X}}$, is combined with the actual values. That is, the missing regions in $\mathbf{X}$ are imputed by the generated values in $\overline{\mathbf{X}}$ from the same location and then sent to D. In this way, $\hat{\mathbf{X}}$ is obtained, and then sent to D with $\mathbf{H}$ to make the decisions

$$\hat{\mathbf{X}} = \mathbf{M} \odot \mathbf{X} + (1 - \mathbf{M}) \odot \overline{\mathbf{X}} \tag{2}$$

Specifically, D wants to maximize the probability of correctly predicting $\mathbf{M}$ whereas G wants to minimize such probability.

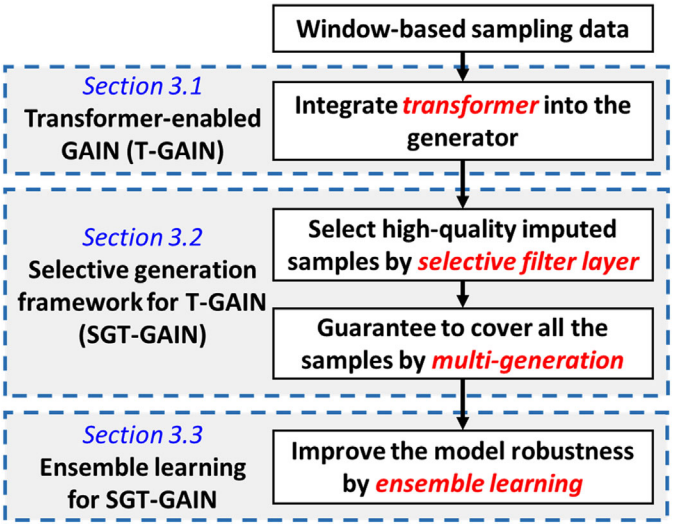

Figure 3. An overview of the proposed SGT-GAIN.

Besides, the prediction of $\mathbf{M}$, i.e., $\hat{\mathbf{M}}$, is $\log\left(D\left(\hat{\mathbf{X}},\mathbf{H}\right)\right)$. Under such circumstances, in order to achieve that the distribution of generated data is similar to that of the distribution of actual data, the minimax game between G and D is demonstrated as the value function in (3). Based on (3), G and D could compete with each other until achieving a good balance:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\bar{\mathbf{X}},\mathbf{M},\mathbf{H}}\left[\mathbf{M}^T \log\left(D\left(\hat{\mathbf{X}},\mathbf{H}\right)\right)\right] + (1-\mathbf{M})^T\left[\log\left(1-D\left(\hat{\mathbf{X}},\mathbf{H}\right)\right)\right] \quad (3)$$

Notably, $\mathbf{M}\odot\overline{\mathbf{X}}$ may be different from $\mathbf{M}\odot\mathbf{X}$, since $\overline{\mathbf{X}}$ only depends on the noise $\mathbf{Z}$. Thus, it is important to increase the similarity between $\mathbf{M}\odot\overline{\mathbf{X}}$ and $\mathbf{M}\odot\mathbf{X}$. Following this direction, the Mean Square Error (MSE) between $\overline{\mathbf{X}}$ and $\mathbf{X}$, i.e., L_M $(\overline{\mathbf{X}},\ \mathbf{X})$, can be calculated and optimized. Hence, the losses for G, L_G and D, L_D are demonstrated in (4) where α is a hyper-parameter:

$$L_D = (1-\mathbf{M})^T\log\left(1-D\left(\hat{\mathbf{X}},\mathbf{H}\right)\right) - \mathbf{M}^T\log\left(D\left(\hat{\mathbf{X}},\mathbf{H}\right)\right)$$

$$L_G = -(1-\mathbf{M})^T\log\left(D\left(\hat{\mathbf{X}},\mathbf{H}\right)\right) + \alpha L_M\left(\overline{\mathbf{X}},\mathbf{X}\right) \quad (4)$$

Due to the inherent properties of the GAN structure, model collapse (Goodfellow *et al.*, 2020) may still occur, leading to an unstable and divergent training process of GAIN. As a result, the GAIN-based estimation of $\mathbf{X}$, i.e., $\overline{\mathbf{X}}$, may be significantly biased. More importantly, the existing GAIN framework cannot capture the sequential effects in the window-based samples. Hence, to address these two limitations in GAIN, a novel transformer-enabled GAIN with selective generation (SGT-GAIN) is proposed in Section 3.

3. Research methodology

This section will introduce the proposed SGT-GAIN to address the limitations of GAIN discussed in Section 2.2. An overview of the proposed SGT-GAIN is shown in Figure 3, which consists of three steps. First of all, to capture the underlying sequential relationships for missing region imputation, the transformer-based neural network is integrated in the generator of GAIN as transformer-enabled GAIN (T-GAIN) (see Section 3.1). Afterwards, to reduce the imputation bias caused by generative adversarial architecture, a novel selective generation mechanism for T-GAIN (SGT-GAIN) is proposed and added into the transformer-enabled generator by incorporating selective filter layer and multi-generation collaboration (see Section 3.2). Then, a bagging-based ensemble learning framework is also applied for SGT-GAIN to further increase the robustness and reduce the imputation variation of the proposed method (see Section 3.3).

3.1 Transformer-enabled GAIN (T-GAIN)

As an emerging AI model, the transformer has been widely applied in various natural language processing tasks; it has also shown its great potential in computer vision areas, due to its strong capability to handle the long-term dependencies in high dimensional data (Gillioz *et al.*, 2020; Tay *et al.*, 2022). Besides, recently Jiang *et al.* (2021) and Zhang *et al.* (2022) also demonstrated that a transformer can make significant contributions to enhancing the capability of the generator in GAN. However, the investigation on the integration of a transformer and GAN are still limited, and this gap has not been completely addressed. In a transformer, one of the most critical components is the multi-head attention mechanism, which considers the pairwise relations among the elements in the input data. Hence, the complex sequential relationships can be learned by the multi-head attention mechanism. Thus, it also motivates this study enabling GAIN to equip the strength of transformer.

An overview of the designed transformer-enabled GAIN is shown in Figure 4(a), in which the transformer is incorporated in the generator. Specifically, in each iteration, the transformer-enabled generator, i.e., G_t, will generate a complete matrix $\overline{\mathbf{X}}$ based on noise $\mathbf{Z}$. Afterwards, the generated $\overline{\mathbf{X}}$ will be combined with $\mathbf{X}$ and $\mathbf{M}$ as $\hat{\mathbf{X}}$ in terms of a similar procedure to that described in (2). Then $\hat{\mathbf{X}}$ and $\mathbf{H}$ will be sent to the discriminator D, which aims to discriminate the area of actual values and imputed values. The output of D, i.e., $D\left(\hat{\mathbf{X}},\mathbf{H}\right)$, will be applied to guide the update of both G_t and D.

Apart from the multi-head attention mechanism, as shown in Figure 4(b), the add & norm module as well as the feed-forward layers are also widely incorporated in a transformer to eliminate the gradient problem and describe nonlinear relationships among the data (Geva *et al.*, 2020). By integrating the above-mentioned components, the transformer incorporated in the generator consists of two main modules: one encoder and one decoder, which have similar structures. With this architecture, the transformer-enabled generator G_t is demonstrated in Figure 4(b).

Specifically, denote one input window-based sample as $\mathbf{X}_{n\times d}$, which has n vectors with d variables. According to Figure 4(b), $\mathbf{X}_{n\times d}$ is first sent into the multi-head attention mechanism in the encoder. Specifically, the multi-head

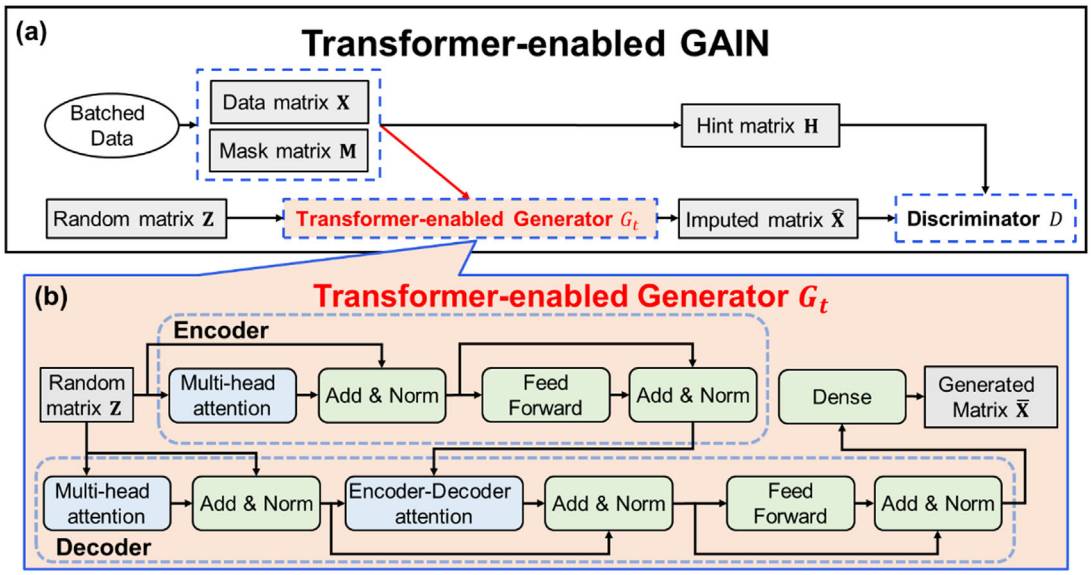

Figure 4. A demonstration of the transformer-enabled GAIN (a) and the transformer-enabled generator (b).

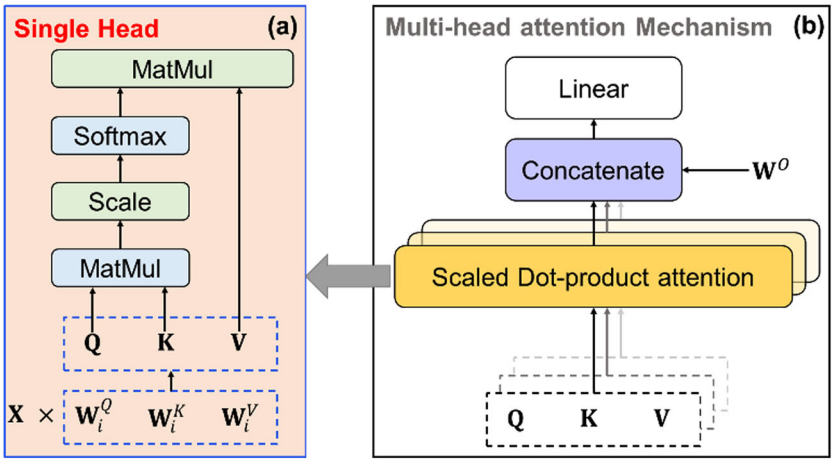

Figure 5. Architecture of the single head (a) and multi-head attention mechanism (b).

attention mechanism consists of several single heads. Suppose that there are h_e single heads in the multi-head attention mechanism. Each single head has a similar structure as shown in Figure 5(a). Three weight matrices, i.e., $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$ and $\mathbf{W}_i^V$, are randomly initialized for the calculation in head i. Each single-head attention, e.g., a head i, could distribute the information (e.g., the area to which the model should pay more attention) in a single direction over the transformer. To improve the ability of a transformer to learn the information from multiple directions, h_e single heads are integrated together as the multi-head attention mechanism in the transformer as shown in Figure 5(b).

Specifically, for head i, $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$ and $\mathbf{W}_i^V$ are used to calculate the mapping elements of queries, keys and values, i.e., $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$, respectively. Afterwards, $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are applied to obtain the output of self-attention mechanism for head i. The equation of the multi-attention mechanism is demonstrated in (5). Finally, the outputs from different heads, i.e., $\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, ..., \mathbf{A}_h\}$, are concatenated together to obtain the final output, i.e., MultiHead$(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, by multiplying another weight matrix $\mathbf{W}^O$ (Tay *et al.*, 2022):

$$\mathbf{Q}_i = \mathbf{X}\mathbf{W}_i^Q \;;\; K_i = \mathbf{X}\mathbf{W}_i^K;\; \mathbf{V}_i = \mathbf{X}\mathbf{W}_i^V$$

$$\mathbf{A}_i = \text{softmax}\left(\frac{\mathbf{Q}_i\mathbf{K}_i^T}{\sqrt{d}}\right)\mathbf{V}_i$$

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_h)\mathbf{W}^O \quad (5)$$

To simplify the model training process and eliminate the gradient problem, the add & norm module is applied as (6). First, the MultiHead$(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ is normalized among different samples by LayerNorm$(\cdot)$. Afterwards, the LayerNorm(Multi Head$(\mathbf{Q}, \mathbf{K}, \mathbf{V})$) is added by the input data matrix to obtain the output, i.e., $\mathbf{X}_\text{A}$:

$$\mathbf{X}_\text{A} = \text{LayerNorm}(\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V})) + \mathbf{X} \quad (6)$$

Then to effectively describe the nonlinear relationships, the feed-forward layer is applied. The input of the feed-forward layer, i.e., $\mathbf{X}_\text{A}$, is passed into a two-layered feed-forward network with ReLU activations, which is expressed as (7), where F_1 and F_2 are functions like $\mathbf{W}\mathbf{x} + \mathbf{b}$. In this way, the output

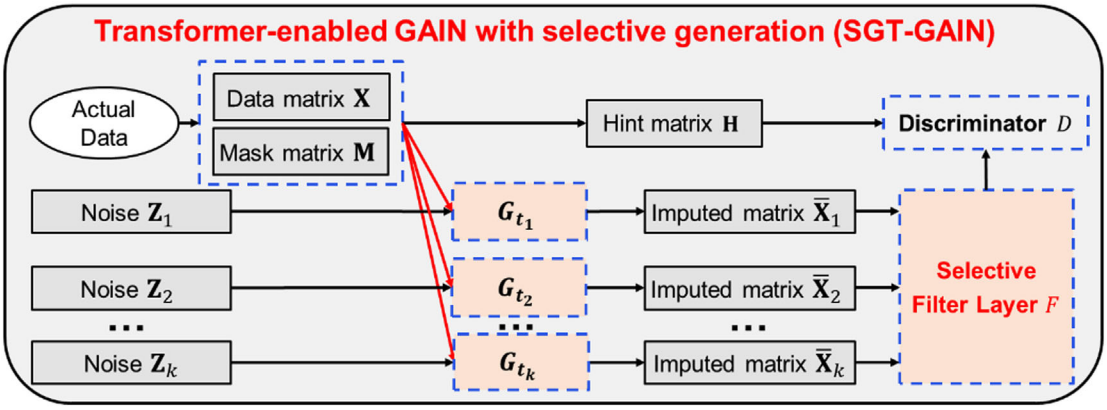

Figure 6. A demonstration of the selective multi-generation framework.

of this encoder, i.e., $\overline{\mathbf{X}}_{\mathrm{A}}$, can be passed either to next encoder or the decoders:

$$\overline{\mathbf{X}}_{\mathrm{A}} = F_2(\mathrm{ReLU}(F_1(\mathbf{X}_{\mathrm{A}}))) \tag{7}$$

Notably, in each encoder, the attention mechanism is self-attention, which means $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$ and $\mathbf{W}_i^K$ are randomly initialized in each encoder. However, in the decoder, it will incorporate its own input and the output of encoders for the prediction (Tay *et al.*, 2022). Hence, as described in Figure 4(b), the decoder has both a self-attention mechanism and an encoder-decoder attention mechanism. For the encoder-decoder attention mechanism, $\mathbf{K}$ and $\mathbf{V}$ are obtained from the last encoder instead of randomly initialization. After the decoders, a dense layer with softmax activation is added to transform the output vectors of the decoder to the vector with desired format. In this way, the pair-wise relationships between each element in the input window-based samples can be learnt. The sequential information among the window-based samples can also be captured for the imputation in the T-GAIN.

The incorporation of transformer in T-GAIN does not change the main adversarial learning architecture. Thus, the convergence property of T-GAIN is still similar to GAN (Goodfellow *et al.*, 2020). Specifically, when the distribution of actual data is similar enough to the generated data, i.e., $P_{\mathbf{data}} = P_{\mathbf{g}}$, the training converges. In addition, although the convergence criteria are still based on the time-independent distribution comparison, the time-dependence pattern can also be well considered, since we adopted the time window-based sampling for the raw time series (each sample is a time-dependent sequence), as discussed in Section 2.1. From this perspective, the time-dependence is mainly considered within each sample instead of between different samples. Consequently, the transformer-based architecture is incorporated to learn the time-dependent pattern in each sample when training the GAN model.

3.2 Enabling selective generation framework for T-GAIN (SGT-GAIN)

In Section 3.1, the transformer is incorporated to learn the global sequential relationships. However, imputation bias may still occur since the values in both the missing and non-missing area are still generated from a single transformer-enabled generator. Hence, to reduce the imputation bias, a novel selective multi-generation framework is proposed. In this way, the T-GAIN with selective generation (SGT-GAIN) is shown in Figure 6. Using this architecture, multiple random matrices are sent to different G_t for selective multi-generation. Afterwards, the imputed matrices are sent to the selective filter layer for selection. In the following sections, the selective filter layer is demonstrated in Section 3.2.1 and the multi-generation framework is discussed in Section 3.2.2.

3.2.1. Selective filter layer

In order to make $\mathbf{M} \odot \overline{\mathbf{X}}$ closer to $\mathbf{M} \odot \mathbf{X}$, inspired by our prior work, the augmented time-regularized GAN (ATR-GAN) (Li *et al.*, 2021), the selective filter layer, F, is proposed in this work as Definition 1 to reduce the imputation bias.

Definition 1. (**Selective filter layer**): Selective filter layer F is proposed to select the generated window-based sample $\overline{\mathbf{X}}$ based on the similarity with $\mathbf{X}$. One-to-one Euclidean distance is calculated among each sample in $\overline{\mathbf{X}}$ and $\mathbf{X}$. With the help of a threshold δ, if the distance of the sample is less than δ, the generated sample will be selected and passed on. Hence, an indicator function I is applied to show which sample to pass on by outputting a 0-1 binary matrix with the same dimension as $\overline{\mathbf{X}}$. The format of the selective filter layer can be mathematically expressed as

$$\overline{\mathbf{X}}^* = F(\overline{\mathbf{X}},\ \mathbf{X}) = \overline{\mathbf{X}} \cdot I_{\left\{d(\overline{\mathbf{X}},\mathbf{X})<\delta\right\}_{n\times c}}(\overline{\mathbf{X}}) \tag{8}$$

It is important to note that, δ is a tuning parameter that can be determined by experimental trials. Specifically, δ can be applied to control the sample size for $\overline{\mathbf{X}}^*$. That is, δ can be set according to the percentile of calculated distance (Li *et al.*, 2021). Hence, there is no neural network parameters in F, which means F does not need to be updated during the training process of the model.

Due to the selection in the selective filter layer, $\overline{\mathbf{X}}^*$ may have the sample size deduction compared with $\mathbf{X}$. Hence, denote that the actual data matrix corresponding to $\overline{\mathbf{X}}^*$ is $\mathbf{X}^*$, and the mask matrix corresponding to $\overline{\mathbf{X}}^*$ is $\mathbf{M}^*$.

Afterwards, $\mathbf{X}^*$ and $\overline{\mathbf{X}}^*$ will be combined with $\overline{\mathbf{X}}^*$ as $\hat{\mathbf{X}}^*$ following (9), and then $\hat{\mathbf{X}}^*$ is sent to the discriminator:

$$\hat{\mathbf{X}}^* = \mathbf{M}^* \odot \mathbf{X}^* + (1 - \mathbf{M}^*) \odot \overline{\mathbf{X}}^* \tag{9}$$

3.2.2 Multi-generator collaboration via a selective generation framework

As described in Section 3.2.1, the data matrix sent to the discriminator is $\hat{\mathbf{X}}^*$ rather than $\hat{\mathbf{X}}$, and the sample size deduction from $\hat{\mathbf{X}}$ to $\hat{\mathbf{X}}^*$ may occur . Then it is possible that some samples in $\mathbf{X}$ may never be sent to the discriminator to make decisions. To address this issue brought from the selective filter layer, the multi-generator collaboration is applied.

Suppose k transformer-enabled generators, $\{G_{t_1}, G_{t_2}, ..., G_{t_k}\}$, are applied to generate artificial samples. Similar to δ in Section 3.2.1, k is also a tuning parameter, which can be determined by experimental trials. As shown in Figure 6, based on the k transformer-enabled generators, k window-based imputed samples, $\{\overline{\mathbf{X}}_1, \overline{\mathbf{X}}_2, ..., \overline{\mathbf{X}}_k\}$, are generated. After passing the imputed samples to the selective filter layer, $\{\overline{\mathbf{X}}_1, \overline{\mathbf{X}}_2, ..., \overline{\mathbf{X}}_k\}$ are updated as $\{\overline{\mathbf{X}}_1^*, \overline{\mathbf{X}}_2^*, ..., \overline{\mathbf{X}}_k^*\}$. Then $\{\overline{\mathbf{X}}_1^*, \overline{\mathbf{X}}_2^*, ..., \overline{\mathbf{X}}_k^*\}$ are combined with the corresponding actual data matrices $\{\mathbf{X}_1^*, \mathbf{X}_2^*, ..., \mathbf{X}_k^*\}$ as $\{\hat{\mathbf{X}}_1^*, \hat{\mathbf{X}}_2^*, \ldots, \hat{\mathbf{X}}_k^*\}$ according to (10):

$$\hat{\mathbf{X}}_i^* = \mathbf{M}_i^* \odot \mathbf{X}_i^* + \left(1 - \mathbf{M}_i^*\right) \odot \overline{\mathbf{X}}_i^* \quad i = 1, 2, ..., k \tag{10}$$

Afterwards, $\{\hat{\mathbf{X}}_1^*, \hat{\mathbf{X}}_2^*, \ldots, \hat{\mathbf{X}}_k^*\}$ are all sent to D to get the losses. The overall structure of losses is similar to the losses described in Section 2.2. As shown in (11), for the ith transformer-enabled generator, it has its own loss, i.e., $L_{G_{t_i}}$, according to the output of D by inputting $\hat{\mathbf{X}}_i^*$. The loss of the discriminator, i.e., L_D, is calculated by utilizing the output of each transformer-enabled generator, i.e., $\{\hat{\mathbf{X}}_1^*, \hat{\mathbf{X}}_2^*, \ldots, \mathbf{X}_k^*\}$. Then $L_{G_{t_i}}$ and L_D could update the entire model accordingly:

$$L_{G_{t_i}} = -\left(1 - \mathbf{M}_i^*\right)^T \log\left(D\left(\hat{\mathbf{X}}_i, \mathbf{H}_i\right)\right) + \alpha L_M(\overline{\mathbf{X}}^*, \mathbf{X}^*)\, i = 1, 2, ..., k$$

$$L_D = \sum_{i=1}^{k} \frac{1}{k} \left(\left(1 - \mathbf{M}_i^*\right)^T \log\left(1 - D(\hat{\mathbf{X}}_i, \mathbf{H}_i)\right) - \mathbf{M}_i^{*T} \log\left(D(\hat{\mathbf{X}}_i, \mathbf{H}_i)\right) \right) \tag{11}$$

When the losses converge, the multi-generator is extracted for data imputation. Then as shown in (12), the overall imputed matrix from SGT-GAIN, i.e., $\hat{\mathbf{X}}$, is obtained by calculating the mean from all the imputed matrices, i.e., $\{\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2, ..., \hat{\mathbf{X}}_k\}$:

$$\hat{\mathbf{X}} = \sum_{i=1}^{k} \frac{1}{k} \hat{\mathbf{X}}_i \tag{12}$$

The overall algorithm for the proposed SGT-GAIN is shown in Algorithm 1. The actual data are first sent to the SGT-GAIN for training. Then the entire data matrix is sent to the transformer-enabled generators to impute the values. The mean of imputed data matrices from transformer-enabled generators are calculated and then output.

Algorithm 1: SGT-GAIN

Input: Actual data matrix for imputation $\mathbf{X}$, Parameter m, k, s and δ

For $j = 1$ **to** k **do**

Step 1: Randomly choose s window-based samples $\mathbf{X}_j$ from actual samples $\mathbf{X}$

Step 2: Generate B artificial samples $\overline{\mathbf{X}}_j$ from transformer-enabled generator $\boldsymbol{G_t}$

Step 3: Send $\overline{\mathbf{X}}_j$ to the selection layer L to obtain $\overline{\mathbf{X}}_j^*$

Step 4: Obtain $\hat{\mathbf{X}}_j^*$ based on $\overline{\mathbf{X}}_j^*$ and $\mathbf{X}_j$

Step 5: Send $\hat{\mathbf{X}}_1^*$, $\hat{\mathbf{X}}_2^*$, ..., $\hat{\mathbf{X}}_k^*$ into discriminator D to get output $D(\hat{\mathbf{X}}_1^*)$, $D\left(\hat{\mathbf{X}}_2^*\right)$, …, $D\left(\hat{\mathbf{X}}_k^*\right)$

Step 6: Optimize the model parameters based on the output of discriminator

Until $L_{G_{t_1}}$, $L_{G_{t_2}}$, …, $L_{G_{t_k}}$ and L_D converge:

Step 7: Send $\mathbf{X}$ to $\{G_{t_1}, G_{t_2}, ..., G_{t_k}\}$ to be imputed as $\hat{\mathbf{X}}$

Output $\hat{\mathbf{X}}$

3.3. Incorporation of ensemble learning framework in SGT-GAIN

Based on the proposed selective generation framework in the transformer-enabled GAIN, the proposed SGT-GAIN can capture the sequential information in the data and impute the data accurately. To further improve the robustness of SGT-GAIN, an ensemble learning framework, which is motivated by bagging (Bühlmann, 2012), is established and incorporated.

As shown in Figure 7, m SGT-GAINs are demonstrated. Similar to δ and k, m is also a tuning parameter which could be determined through experimental trials. To learn the actual data distribution more comprehensively, each SGT-GAIN could have a different concentration. That is, m data matrices, $\{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_m\}$, are obtained from the data tensor $\boldsymbol{\mathcal{X}}$ through bootstrapping, and then sent to the m SGT-GAINs separately. In addition, to emphasize the different concentration of different SGT-GAIN, each SGT-GAIN may apply different hint rates h.

After the SGT-GAINs are well-trained, the entire data tensor $\boldsymbol{\mathcal{X}}$ is sent into each SGT-GAIN to obtain the imputed tensors, i.e., $\{\hat{\boldsymbol{\mathcal{X}}}_1, \hat{\boldsymbol{\mathcal{X}}}_2, ..., \hat{\boldsymbol{\mathcal{X}}}_m\}$. Notably, the continuous and discrete elements are considered separately for imputation. For the continuous elements, the mean from all the imputed data tensors is calculated while the median is calculated for the discrete elements. Then the final imputed data tensors, i.e., $\boldsymbol{\mathcal{X}}'$, can be obtained. In this way, though some SGT-GAINs may provide inappropriate imputed values, such as outliers, due to the different concentrations, the effects on $\boldsymbol{\mathcal{X}}'$ will be significantly eliminated.

Notably, if the distribution of the training set for the proposed method is different from the distribution of the data

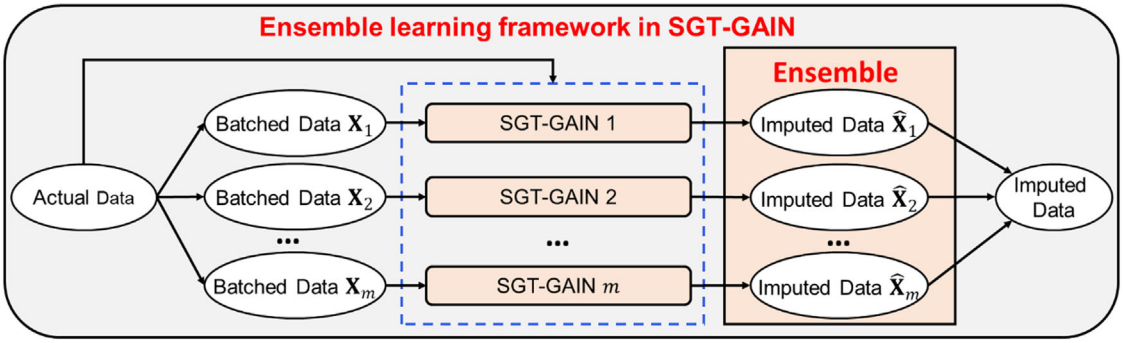

Figure 7. The overview for incorporation of ensemble learning framework in SGT-GAIN.

for imputation, the imputation performance may become worse. However, in practice, the SGT-GAIN model can also be updated to improve the imputation performance when new data arrive.

4. Case studies

In this section, two studies were conducted to validate the effectiveness of the proposed SGT-GAIN. The numerical simulation study is presented in Section 4.1, followed by a real-world case study based on an AM platform, which is discussed in Section 4.2. Specifically, to demonstrate the effectiveness of the proposed method, four benchmark approaches are applied for comparison. They are GAIN (Yoon *et al.*, 2018), T-GAIN (no selective generation and ensemble learning), ensemble GAIN (E-GAIN, no transformer-enabled generator and selective generation), ensemble selective multi-generation GAIN (ESM-GAIN (Li *et al.*, 2022), no transformer-enabled generator) and SGT-GAIN (o ensemble) (SGT-GAIN without ensemble learning framework). In addition, to fully show the capability of the proposed method under different level of missing regions, all the experiments are conducted under different missing region size. The missing region size is personalized according to different studies to fit the actual data.

4.1 Simulation study

To generate data with clear sequential effects, the Gaussian process is applied to simulate 1000 actual samples with 300 sequential points based on the Radial Basis Function (RBF) kernel, i.e., $n = 300$, $d = 1$. The detailed process to simulate the data is shown in (13). The parameter θ in the RBF kernel is set as 0.001. Besides, to make the simulation data more closely match the data from real-world cases, some random Gaussian noises $\mathbf{Z}_{GP}$ are also added:

$$\mathbf{X} = \mathbf{X}_{GP} + \mathbf{Z}_{GP}, \mathbf{X}_{GP} = \begin{bmatrix} \mathbf{x}_1 \\ \cdots \\ \mathbf{x}_{1000} \end{bmatrix}, \mathbf{Z}_{GP} = \begin{bmatrix} \mathbf{z}_1 \\ \cdots \\ \mathbf{z}_{1000} \end{bmatrix},$$

$$\mathbf{x_i} \sim GP(0, \kappa), \; z_i \sim N(0, \; 2^2),$$

$$\kappa(x_{il_1}, x_{il_2}) = \exp\left(-\frac{1}{2\theta}\|x_{il_1} - x_{il_2}\|_2^2\right), \theta = 0.001$$

$$i = 1, 2, \ldots, 1000, \; l_1, l_2 = 1, 2, \ldots, 300 \tag{13}$$

In this way, the $1000 \times 300 \times 1$ data matrix $\mathbf{X}_W$ is generated. Recalling the notation introduced in Section 2.1, the missing region size is $r \times d$ and the first missing point of the ith window-based sample is the f_i-th point. In this study, f_i is randomly determined in the ith sample. Afterwards, r successive points in the samples are removed to simulate the missing regions. To demonstrate the effectiveness of the proposed method comprehensively, experiments are conducted under r from $\{100, \; 120, \; 140, \; 160, \; 180\}$ out of 300.

The imputation performance can be evaluated according to the quality of the imputed regions. Thus, the Mean Absolute Errors (MAEs) between the imputed region and the actual values of the missing region can be used as the evaluation metric to quantitative described the similarity between the ground truth and imputation. It is natural that the lower MAEs indicate better imputation performance.

4.1.1 Parameter selection

In the proposed method, there are three key hyper-parameters, i.e., δ, k, m. Thus, it is important to discuss their influence on the model performance as well as the selection. In this work, δ is set as the percentile of the calculated distance instead of a specified threshold. Since a batch of the samples is sent to the selective filter layer, the number of samples passing the selective filter layer is fixed and can be controlled by δ. Hence, instead of the sample size, δ may influence the selection of k to fully cover all the samples. Hence, the discussion of these three hyper-parameters can be categorized by two groups: m and the group of δ and k. In the experiment, the performance evaluation is based on the MAE and the missing region size is set as 120 for parameter tuning.

Different values of m, including $m = 1, 5, 10, 20$, are considered for selection. Specifically, under each value of m, four different pairs of δ and k are also selected, $\{\delta = 100\%, \; k = 1\}$, $\{\delta = 80\%, \; k = 2\}$, $\{\delta = 60\%, \; k = 5\}$, $\{\delta = 40\%, \; k = 10\}$, to fully investigate the importance of m. The MAEs of the proposed method under different pairs of $\{\delta, k\}$ and m are shown in Table 1. Under each pair of $\{\delta, k\}$, it can be observed that the proposed method always has the smallest MAE when $m = 5$. In addition, the MAE when $m = 1$ is always higher than the others under each pair of $\{\delta, k\}$, which also shows that the ensemble learning framework can help to

Table 1. The MAEs under different m and pairs of $\{\delta, k\}$.

	m			
$\{\delta, k\}$	*1*	*5*	*10*	*20*
$\{\delta = 100\%,\ k = 1\}$	2.17	2.06	2.12	2.15
$\{\delta = 80\%,\ k = 2\}$	2.06	**1.93**	2.00	1.99
$\{\delta = 60\%,\ k = 5\}$	2.13	2.01	2.07	2.07
$\{\delta = 40\%,\ k = 10\}$	2.23	2.10	2.17	2.16

Table 2. The MAEs under different pairs of $\{\delta, k\}$.

	δ			
k	*40%*	*60%*	*80%*	*100%*
1	3.46	2.39	1.96	2.06
2	2.46	2.01	**1.93**	1.94
5	2.27	2.01	2.05	2.12
10	2.10	2.00	2.39	2.07

improve the performance of the proposed method. Also, when m is 10 or 20 (i.e., relatively large), the MAEs of the proposed method are similar under all of the four $\{\delta, k\}$ pairs. Hence, the improvement of the proposed method through tuning δ and k may be insignificant when m is relatively large. Therefore, m is chosen to have a value of five in this work.

For the parameters δ and k, each pair of $\{\delta, k\}$ from $\delta = 40\%, 60\%, 80\%, 100\%$ and $k = 1, 2, 5, 10$ are selected. Specifically, $\delta = 100\%$ means the selective filter layer does not work. The MAEs of the proposed method are shown in Table 2. Under each specific k, the MAEs when $\delta = 100\%$ are always not the smallest, which shows that the selective filter layer can be helpful to impute the proposed method more accurately. In addition, when $\delta = 40\%$ and $k = 1$, the MAE of the proposed method is 3.46, which is much higher than the other MAEs. This is due to the relatively small δ and the only generator, which is too hard to train the model accurately. In addition, the optimal values of δ and k should be negatively correlated: when $\delta = 40\%$ or 60%, $k = 10$ may get the smallest MAEs. On the other hand, $k = 2$ may get the smallest MAEs when $\delta = 80\%$ or 100%. Especially when $\delta = 80\%$ and $k = 20$, it is clearly shown that the proposed method has the smallest MAE. Hence, the pair of $\delta = 80\%$ and $k = 2$ is selected in this study.

In practice, as k and m increase, the model complexity may also significant increase, leading to a higher training time. For instance, the training time may increase by 1.5 seconds for each k and each m. However, it will not influence the application of the SGT-GAIN model. A well-trained SGT-GAIN model can always impute about 20 window-based samples within a second. Hence, the imputation frequency could be mostly higher than the sampling frequency in real-world applications.

The detailed setups of parameters in this study are shown in Table 3. Notably, each multi-head attention mechanism of the transformer-enabled generator in each SGT-GAIN involves two heads. The detailed structure of the employed transformer follows the description in Section 3.1. As for the discriminator in each GAIN-based model, they consist of a three-layer MultiLayer Perceptron (MLP). The first two layers utilizes ReLu as activation functions while the last layer applies sigmoid as the activation function. The total training time for the proposed model is about 15 mins.

Table 3. The data and parameter setups.

Setup	*Value*
Sample size	$1000 \times 300 \times 1$
Number of transformer-enabled generators k	2
Number of SGT-GAIN m	5
Threshold δ	80th percentile of the calculated distance

4.1.2. Missing region size-based discussion

To fully validate the performance of the proposed SGT-GAIN, multiple benchmark comparisons as well as ablation experiments are performed. Yoon *et al.* (2018) have demonstrated the effectiveness of GAIN with some conventional and machine learning imputation approaches such as matrix completion (Mazumder *et al.*, 2010), KNN (Zhang, 2012), and MissForest (Stekhoven and Bühlmann, 2012). Hence, this work will focus on the comparison between the proposed method and the GAIN-based approaches. Thus, GAIN (Yoon *et al.*, 2018), T-GAIN (no selective generation and ensemble learning), ensemble GAIN (E-GAIN, no transformer-enabled generator and selective generation), ensemble selective multi-generation GAIN (ESM-GAIN, no transformer-enabled generator (Li *et al.*, 2022)), and SGT-GAIN (o ensemble) (the proposed method without ensemble learning framework) are applied as benchmark approaches. To ensure the fairness of comparison, the above-mentioned benchmarks will have the same parameter setup as SGT-GAIN. Particularly, the generators in GAIN, E-GAIN and ESM-GAIN are also a three-layer MLP similar to the discriminators. In addition, to make the results more representational, each experiment involves five replicates and then the average MAEs (with standard deviation) are used for comparison.

The MAEs between the proposed method and benchmark approaches under different missing region size are shown in Figure 8(a) and Figure 8(b). Specifically, to better show the MAE differences between the proposed method and benchmark approaches, the comparisons are divided into two groups. As shown in Figure 8(a), the group 1 is comparing transformer-enabled GAIN models, i.e., SGT-GAIN, T-GAIN and SGT-GAIN (o ensemble), to demonstrate the effectiveness of transformer architecture and ensemble learning framework. As for the group 2, it compares the SGT-GAIN with other benchmark approaches, i.e., GAIN, E-GAIN and ESM-GAIN, to demonstrate the effectiveness of the incorporation of selective generation framework and ensemble learning framework.

As shown in Figure 8(a), the MAEs of the proposed method are lower than the MAEs of T-GAIN when the missing region size is 100, 120 and 160. When the missing region size is 140 and 180, the MAEs of SGT-GAIN still do not exceed the MAEs of T-GAIN. Hence, the MAEs comparison between the proposed method and T-GAIN can show the effectiveness of the incorporation of selective

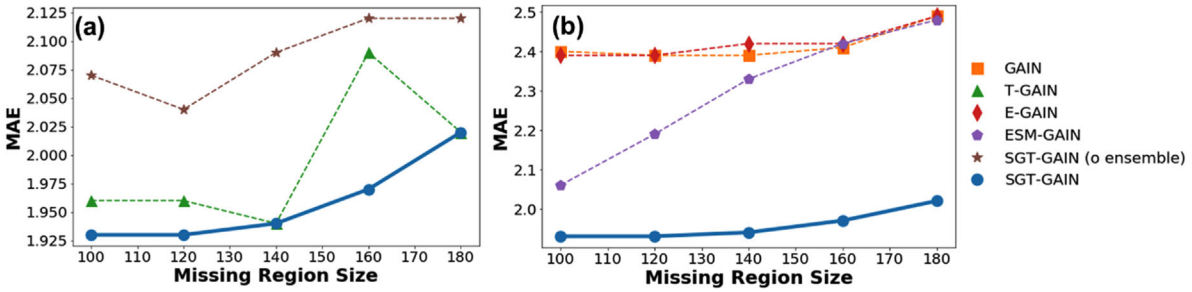

Figure 8. MAE comparisons between the proposed method and benchmark approaches: (a) Comparisons between SGT-GAIN, T-GAIN and SGT-GAIN (o ensemble); (b) comparisons between SGT-GAIN, GAIN, E-GAIN and ESM-GAIN.

Table 4. The MAEs and standard deviations under different missing region size.

	Missing region size				
Approaches	*100*	*120*	*140*	*160*	*180*
GAIN	2.40 (0.02)	2.39 (0.03)	2.39 (0.04)	2.41 (0.04)	2.49 (0.11)
T-GAIN	1.96 (0.01)	1.96 (0.03)	1.94 (0.04)	2.09 (0.14)	2.02 (0.06)
E-GAIN	2.39 (0.02)	2.39 (0.04)	2.42 (0.05)	2.42 (0.03)	2.49 (0.04)
ESM-GAIN	2.06 (0.02)	2.19 (0.04)	2.33 (0.18)	2.42 (0.03)	2.48(0.06)
SGT-GAIN (o ensemble)	2.07 (0.02)	2.04 (0.04)	2.09 (0.05)	2.12 (0.13)	2.12 (0.05)
SGT-GAIN	**1.93 (0.02)**	**1.93 (0.02)**	**1.94 (0.03)**	**1.97 (0.04)**	**2.02 (0.04)**

generation framework and ensemble learning framework. Besides, the MAEs of SGT-GAIN are always smaller than the MAEs of SGT-GAIN (o ensemble), which also demonstrates the effectiveness of the ensemble learning framework. In addition, as the missing region size increases, the MAEs of the proposed SGT-GAIN also increase smoothly whereas the MAEs of the other two benchmark approaches increase non-smoothly, indicating potential low robustness and higher variation. Besides, the MAEs of SGT-GAIN (o ensemble) are also smoother than the MAEs of T-GAIN. Hence, it also proves the effectiveness of the incorporation of the selective generation framework and ensemble learning framework to improve model robustness, especially the ensemble learning framework. Notably, such increasing patterns of MAEs are reasonable, since higher missing region size means less information in the data to be learnt for imputation.

Comparing the proposed method with ESM-GAIN, as shown in Figure 8(b), the proposed SGT-GAIN still has the smaller MAEs. Hence, such MAE differences demonstrate that the transformer-enabled generator is very effective to handle the complex sequential effects. Specifically, as the missing region size increases, the MAEs differences between SGT-GAIN and ESM-GAIN increases rapidly. It also successfully proves that the transformer-enabled generator could work much more stable than the MLP generator better when the missing region is large.

Besides, as the missing region size is 100, 120 and 140, the MAEs of ESM-GAIN are smaller than E-GAIN, which could clearly demonstrate the effectiveness of the proposed selective generation framework. As the missing region size becomes larger, i.e., 160 and 180, the MAEs of ESM-GAIN and E-GAIN are very similar. Therefore, it shows that the performance of the selective generation framework is limited when the missing region size is relatively large, i.e., any region that consists of more than half of the points in the window being missing. Furthermore, as the missing region size increases, the MAEs for the approaches in group 2 also increase, which also fits the recognition that the missing region imputation task becomes more difficult.

The MAEs of the SGT-GAN are also lower than the MAEs of GAIN and E-GAIN. However, the MAEs of GAIN and E-GAIN are very similar, so that it is hard to demonstrate the effectiveness of the ensemble learning framework. The main reason is that effective ensemble learning depends on the overall performance of the learners, but both GAIN and E-GAIN do not learn the sequential effects very well. In addition, since the main goal of the ensemble learning framework is to reduce the imputation variations, the means and standard deviations of MAEs under different missing region size are provided in Table 4 for a better comparison. Compared with GAIN, the standard deviations of E-GAIN are mostly smaller especially when the missing region size is 180. Hence, it could validate that the ensemble learning framework is able to improve the model robustness. Besides, the standard deviations of the proposed method are mostly the lowest, which also shows the high robustness of the proposed method. As the missing region size increases, the standard deviations of all the approaches gradually increase, which also proves that the imputation task for GAIN-based models gradually become more difficult. Overall, the simulation study demonstrates the superior performance of the proposed SGT-GAIN.

In addition, to further demonstrate the imputation accuracy, more evaluation metrics are leveraged, including the relative MAEs as well as the correlation coefficients between the imputed data and actual data. The relative MAEs are the MAEs in percentage to quantify the imputation bias. To quantify the correlation between the imputed data and actual data, the Pearson correlation coefficient (Cohen *et al.*, 2009) is applied. The results of relative MAEs and the correlation coefficients comparison are shown in Table 5. It demonstrates that the proposed method has the smallest relative MAEs and the highest correlation coefficients under different missing region sizes. Therefore, both relative MAEs and correlations also demonstrate the outperformance of the proposed method, which is consistent with the comparison results using MAE.

Table 5. Relative MAEs and correlation (in brackets) under different missing region size.

	Missing region size				
Approaches	*100*	*120*	*140*	*160*	*180*
GAIN	12.00% (0.55)	11.95% (0.54)	11.95% (0.56)	12.05% (0.56)	12.45% (0.534)
T-GAIN	9.80% (0.67)	9.80% (0.66)	9.70% (0.69)	10.45% (0.64)	10.10% (0.658)
E-GAIN	11.95% (0.55)	11.95% (0.54)	12.10% (0.55)	12.10% (0.56)	12.45% (0.536)
ESM-GAIN	10.30% (0.64)	10.95% (0.59)	11.65% (0.58)	12.10% (0.56)	12.40% (0.536)
SGT-GAIN (o ensemble)	10.35% (0.64)	10.20% (0.64)	10.45% (0.64)	10.60% (0.64)	10.60% (0.627)
SGT-GAIN	9.65% (0.68)	9.65% (0.67)	9.70% (0.69)	9.85% (0.68)	10.10% (0.658)

Table 6. MAEs under different d and different missing region size.

		Missing region size r				
Dimension d	*Methods*	*100*	*120*	*140*	*160*	*180*
1	GAIN	2.4	2.39	2.39	2.41	2.49
	T-GAIN	1.96	1.96	1.94	2.09	2.02
	E-GAIN	2.39	2.39	2.42	2.42	2.49
	ESM-GAIN	2.06	2.19	2.33	2.42	2.48
	SGT-GAIN (o ensemble)	2.08	2.04	2.09	2.12	2.12
	SGT-GAIN	**1.93**	**1.93**	**1.94**	**1.97**	**2.02**
3	GAIN	2.16	2.17	2.19	2.23	2.69
	T-GAIN	2.03	2.04	2.10	1.93	2.00
	E-GAIN	2.17	2.21	2.29	2.17	2.26
	ESM-GAIN	2.20	2.18	2.25	2.24	2.67
	SGT-GAIN (o ensemble)	2.05	2.05	2.22	2.14	2.37
	SGT-GAIN	**1.87**	**2.01**	**2.09**	**1.93**	**2.00**
5	GAIN	2.06	2.03	2.22	2.11	3.17
	T-GAIN	1.95	1.96	1.97	2.02	2.05
	E-GAIN	2.04	2.07	2.08	2.17	2.20
	ESM-GAIN	2.05	2.00	2.08	2.21	2.16
	SGT-GAIN (o ensemble)	1.99	2.16	2.09	2.08	2.13
	SGT-GAIN	**2.07**	**1.90**	**2.03**	**1.92**	**1.91**
10	GAIN	1.92	1.98	1.94	2.05	2.26
	T-GAIN	2.01	2.13	2.01	2.08	2.24
	E-GAIN	1.89	1.95	1.86	2.02	2.35
	ESM-GAIN	1.92	1.95	1.99	2.20	2.09
	SGT-GAIN (o ensemble)	2.08	2.17	2.11	2.18	2.23
	SGT-GAIN	**2.01**	**2.16**	**1.91**	**1.97**	**2.07**

4.1.3 Autocorrelation-based discussion

In the previous experiments, the experiments were conducted for $d = 1$, which means the autocorrelation in the actual data is limited. Therefore, to further demonstrate the effectiveness of the proposed method under different levels of autocorrelation, experiments under two aspects are conducted: (i) experiments under different d, i.e., experiments under different data dimensions; (ii) experiments under different θ, i.e., experiments where the distributions of training set and testing set are different.

In the first group of experiments, four values of d, including $d = 1, 3, 5, 10$, are applied to demonstrate the effectiveness of the proposed method where the other setups remain the same. The recorded MAEs are shown in Table 6. Under each r and each d, the smallest MAEs are highlighted. According to the results, the proposed method has the smallest MAEs under each r when $d = 1$ and $d = 3$. However, as $d = 5$ or $d = 10$, the proposed method can achieve the smallest MAEs when r is relatively large, i.e., $r = 160$ or 180. When the dimension is high, the larger missing region size means less available information for imputation. Hence, the proposed method is more competitive when the imputation task is more complicated. Overall, the proposed method has the potential to be applied when data dimensionality is high.

Table 7. The MAEs and standard deviations (in brackets) under different θ of SGT-GAIN.

	Missing region size				
θ	*100*	*120*	*140*	*160*	*180*
0.001	1.93 (0.02)	1.93 (0.02)	1.94 (0.03)	1.97 (0.04)	2.02 (0.04)
0.002	2.03 (0.08)	2.34 (0.04)	2.33 (0.25)	2.11 (0.00)	2.21 (0.00)
0.003	2.22 (0.22)	2.24 (0.11)	2.44 (0.01)	2.45 (0.10)	2.34 (0.15)
0.005	2.13 (0.03)	2.12 (0.13)	2.27 (0.13)	2.25 (0.04)	2.41(0.26)

To demonstrate the capability of the proposed method under different operation conditions, i.e., when the distributions of the training set and the testing set are slightly different, more experiments are conducted with $d = 1$. The proposed method is initially trained by the actual data simulated for $\theta = 0.001$, and then it is tested by the actual data simulated under $\theta = 0.002$, 0.003 or 0.005. The MAEs of the proposed method and its standard deviations (in brackets) under different missing region sizes are shown in Table 7. When θ increases, the MAEs of the proposed method will also increase, since higher θ means the higher bias from the training set to the testing set, leading to higher difficulty to impute the missing data. However, as shown in Table 7, the standard deviations may vary a lot without any discernable pattern, which means the difference of the distributions may increase the variation of imputation. Overall, the MAEs of the proposed method may increase at most 25%, which is much less than the relative increment of θ (500%). Therefore, the proposed method still has the application potential when the operation conditions slightly change.

4.2 Real-world case study in AM

In this section, a real-world AM case study for online anomaly detection is conducted. Due to design mechanism of AM, the product is printed layer-by-layer (Sturm *et al.*, 2017). However, it is possible that the printed product may have defects or unintended anomalies (Liu *et al.*, 2020), which may lead to labor and financial costs. The change of geometric structure due to unintended changes could be reflected in the online sensor signals (Shi et al., 2023), i.e., the collected time series sensor data. Hence, online anomaly detection could be performed via online sensor data. However, as discussed in Section 1, due to the potential data loss issue during the data transmission, it is important to address the missing region issue, so that the detection could be more effective. In this study, the data collection and the experimental setup for missing region imputation are introduced in Section 4.2.1, followed by the results and discussions in Section 4.2.2.

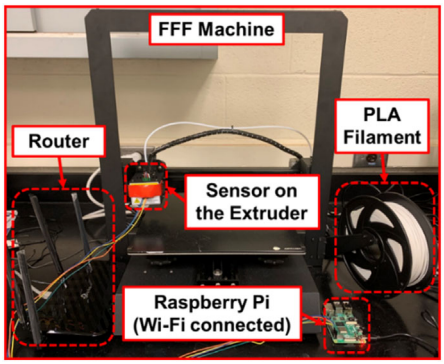

Figure 9. Experimental platform setup.

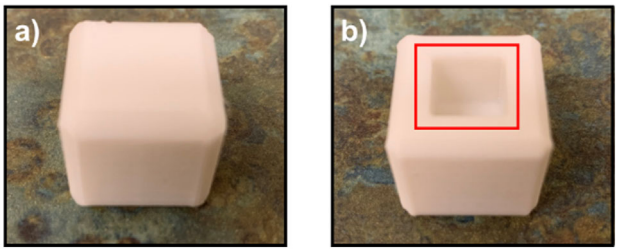

Figure 10. Sample cubes with normal (a) and anomaly (part within red solid line) (b).

4.2.1 Data collection and experimental setup

In this case, as shown in Figure 9, the data was collected from one accelerometer attached on the extruder of a regular fused filament fabrication (FFF) machine. The installed accelerometer has three channels, and the sampling frequency was approximately 1 Hz. The Raspberry Pi 4b microcontroller was used for data acquisition from the accelerometer. During each printing, there is a total of 1927 sensor signals collected by time. Without loss of generality, a solid cube was printed using polylactic acid (PLA) filament as shown in Figure 9. The dimension of the cube is $2 \times 2 \times 2\,\text{cm}^3$.

To simulate the anomaly, compared with the normal part (Figure 10(a)), a small square void was intentionally inserted into the cube within the red solid line as shown in Figure 10(b). In this way, when the layers are printed without the square void, the vibration signals are collected under normal state. When the layers are printed in the layer with the square void, the vibration signals are collected under an abnormal state (Shi *et al.*, 2022). Approximately, the first 40% of observations are collected when the layers are printed without the void, while the remaining 60% are collected when the layers are printed in the layer with the void. Hence, in this case, the first 5–30% of the collected data is extracted as normal samples, whereas the latter 60–90% of the collected data is extracted as abnormal samples. In addition, the window size, i.e., n, is set as 50. In order to increase the number of window-based samples, the overlap size between adjacent windows is 40. Then the number of samples is demonstrated in Table 8.

If the proposed SGT-GAIN method can provide effective imputation, the trained anomaly detector, i.e., a classification model, should classify the imputed samples accurately. Thus, the performance of the proposed method could be justified by comparing the anomaly detection results after imputing missing regions with benchmark approaches, i.e., the comparison of classification accuracy. According to the experimental setup, 80% window-based samples are randomly selected as the training set for anomaly detection while the other 20% window-based samples are considered as the testing set. In addition, in each window-based sample of the testing set, some regions are randomly removed as missing regions as described in Section 2.1. Since the dimensions of window-based samples are 50×3, the missing region size, i.e., r, is selected from $\{10, 15, 20, 25, 30, 35, 40\}$ out of 50, as shown in Table 3. The other experimental setups of the proposed method in this case are the same as Section 4.1, and the same benchmark approaches are applied for comparison.

Table 8. The information on the collected data and the setup of computational experiments.

Setup	*Size*
Sample size	578 normal observations, 578 abnormal observations
Anomaly detection training set	924 observations
Anomaly detection testing set	232 observations
Missing region size	$\{10, 15, 20, 25, 30, 35, 40\}$

Notably, without the loss of representativeness, the commonly used gradient boosting classifier (Friedman, 2002) is selected as the anomaly detector based on our preliminary detector comparisons. In addition, the F-score (Sasaki, 2007), which is commonly used in the evaluation of classification performance, is considered as the metric to measure the performance of the proposed method. The baseline F-score of the anomaly detection in this work, i.e., the F-score when the testing set does not need to perform the imputation, is 0.751.

4.2.2 Results and discussions

Similar to Section 4.1, the benchmark approaches are divided into two groups for comparisons. The F-score comparisons between the SGT-GAIN, T-GAIN and SGT-GAIN (o ensemble) are shown in Figure 11(a). The F-scores of the proposed method are higher than T-GAIN under each missing region size, which demonstrates the effectiveness of the incorporation of selective generation framework and ensemble learning framework. In addition, the F-scores of SGT-GAIN (o ensemble) are also mostly higher than T-GAIN and lower than SGT-GAIN. Hence, it also demonstrates the effectiveness of ensemble learning framework. Particularly, when the missing region size is larger than 25, the F-scores of all three approaches are very similar. Hence, it also proves that the incorporation of selective generation framework and ensemble learning framework is limited when the missing region size is more than half of the window size.

As shown in Figure 11(b), the SGT-GAIN still has the highest F-scores under all the missing region size. Therefore, comparing SGT-GAIN with ESM-GAIN, the F-score improvement due to the transformer-enabled generator is significant. In addition, the F-score differences between SGT-GAIN and other GAIN-based approaches in group 2 become larger when increasing the missing size. It also proves that the transformer-enabled generator is more robust and effective, which means the proposed method is more effective than the benchmark approaches.

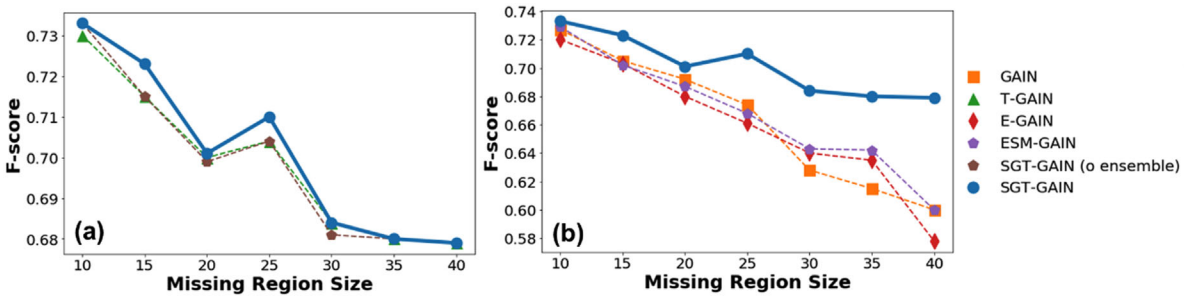

Figure 11. F-score comparisons between the proposed method and benchmark approaches: (a) Comparisons between SGT-GAIN, T-GAIN and SGT-GAIN (o-ensemble); (b) comparisons between SGT-GAIN, GAIN, E-GAIN and ESM-GAIN.

Table 9. The F-scores and standard deviations under different missing region size.

	Missing region size						
Approaches	*10*	*15*	*20*	*25*	*30*	*35*	*40*
GAIN	0.727 (0.011)	0.705 (0.014)	0.692 (0.014)	0.674 (0.032)	0.628 (0.028)	0.615 (0.033)	0.600 (0.067)
T-GAIN	0.730 (0.009)	0.715 (0.011)	0.700 (0.008)	0.704 (0.005)	0.684 (0.011)	0.680 (0.006)	0.679 (0.012)
E-GAIN	0.720 (0.012)	0.703 (0.011)	0.680 (0.015)	0.661 (0.024)	0.640 (0.031)	0.635 (0.034)	0.578 (0.039)
ESM-GAIN	0.729 (0.014)	0.702 (0.012)	0.687 (0.014)	0.668 (0.029)	0.643 (0.031)	0.642 (0.018)	0.600 (0.035)
SGT-GAIN (o ensemble)	0.733 (0.009)	0.715 (0.011)	0.699 (0.008)	0.704 (0.005)	0.681 (0.011)	0.680 (0.009)	0.679 (0.010)
SGT-GAIN	**0.733 (0.008)**	**0.723 (0.005)**	**0.701 (0.004)**	**0.710 (0.007)**	**0.684 (0.005)**	**0.680 (0.008)**	**0.679 (0.010)**

Besides, the F-scores of ESM-GAIN are also mostly higher than both GAIN and E-GAIN, which means the selective generation framework is also effective for missing region imputation. Furthermore, as the missing region size increases, the F-scores for the approaches in both group 1 and group 2 also decrease. Such pattern also proves that higher missing region size lead to less information for missing region imputation, resulting in lower F-scores.

Moreover, as shown in Figure 11(b), the F-scores of E-GAIN are still similar to the F-scores of GAIN, which means the ensemble learning framework does not significantly contribute to improving the imputation accuracy in this study. Since the ensemble learning framework is applied to reduce the imputation variation, the variation comparisons of F-scores should be considered. Then the means and standard deviations of F-scores are shown in Table 9.

As described in Table 9, the standard deviations of the proposed method are mostly the lowest. Particularly, compared with ESM-GAIN, the standard deviations of SGT-GAIN mostly decrease by more than 50%, which shows the high robustness of the proposed method. Compared with GAIN, the standard deviations of E-GAIN are also mostly smaller, which validates that the ensemble learning framework could improve the model robustness. All the pattern descriptions in this section are consistent with the descriptions in Section 4.1. Therefore, the real-world case study in AM also demonstrates the superior performance of the proposed SGT-GAIN for missing region imputation. Moreover, the proposed method can be directly applied to the in-process anomaly detection.

Regarding its potential on real-time applications, in this study, the trained SGT-GAIN model can impute about 20 window-based samples within a second (i.e., about 20 Hz) in this case (by Python 3.7.4 on Intel® Core™ Processor i7-9750H (Hexa-Core, 2.60 GHz)). Therefore, compared to the sampling frequency (1 Hz), the computation efficiency of the proposed method is sufficient enough for online anomaly detection.

5 Conclusions

In this article, a new data imputation approach termed transformer-enabled GAIN with selective generation (SGT-GAIN) is proposed to address the critical missing region issue. The main contributions of the proposed SGT-GAIN consist of three aspects: (i) a transformer-enabled generator is demonstrated to capture the sequential relationship among the window-based samples; (ii) the selective generation framework is proposed to reduce the imputation bias and learn the data patterns comprehensively; and (iii) the ensemble learning framework is incorporated with SGT-GAIN to improve the model robustness and reduce the imputation variation.

The outperformance of SGT-GAIN over the benchmark approaches is demonstrated in the numerical simulation and a real-world case study in AM. In the simulation study, the proposed method has the smallest MAEs, which shows that the proposed method could impute the missing regions accurately. In the real-word AM case study, the proposed method also has the highest F-scores under different missing region size, which also shows the high robustness and accuracy of the proposed method for missing region imputation. In addition, the effectiveness of the components in the proposed SGT-GAIN is also validated in both simulation study and a real-world case study in AM. Thus, the proposed method is very promising for missing region imputation. Specifically, both the simulation study and the real-world case study are conducted mostly when the operation conditions for both training set and the testing set are the same. Hence, in the future, more experiments will be conducted to further investigate the capability of this method when the training set and the testing set follow different distributions. In addition, it is valuable to consider the convergence criteria based on time-dependent distribution comparisons in the future work.

Funding

This work is partially supported by the National Science Foundation under Award Number IIP-2141184.

Notes on contributors

Yuxuan Li received a BS degree in statistics from Renmin University of China, Beijing, China, in 2019. He is currently pursuing a PhD degree in industrial engineering and management at Oklahoma State University, Stillwater, OK, USA. His current research focuses on the advanced data analytics in smart manufacturing and healthcare systems. He is a member of IISE, INFORMS and IEEE.

Zhangyue Shi received a BS degree in mechanical engineering from Xian Jiaotong University, Xian, China, in 2019. He is currently pursuing a PhD degree in industrial engineering and Management with a minor in statistics at Oklahoma State University, Stillwater, OK. His current research interest includes advanced data analytics-based quality assurance in smart manufacturing.

Chenang Liu received his double BS degrees in Environmental and Resource Sciences and Mathematics from Zhejiang University, China, in 2014; he then earned his MS degree in Statistics and PhD degree in Industrial and Systems Engineering from Virginia Tech in 2017 and 2019, respectively. He is currently an assistant professor in the School of Industrial Engineering and Management at Oklahoma State University. His research interests include data-driven analytics and machine learning-enabled modeling to advance smart manufacturing, healthcare, and service systems, as well as applied artificial intelligence for engineering applications.

References

Bühlmann, P. (2012) *Bagging, Boosting and Ensemble Methods*, Springer, Berlin, Germany.

Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., Huang, Y. and Cohen, I. (2009) Pearson correlation coefficient. *Noise Reduction in Speech Processing*, Springer-Verlag, Berlin, Germany, pp. 1–4.

Dogan, A., Li, Y., Odo, C.P., Sonawane, K., Lin, Y. and Liu, C. (2023) A utility-based machine learning-driven personalized lifestyle recommendation for cardiovascular disease prevention. *Journal of Biomedical Informatics* 141, 104342.

Friedman, J.H. (2002) Stochastic gradient boosting. *Computational Statistics & Data Analysis*, **38**(4), 367–378.

García-Laencina, P.J., Sancho-Gómez, J.-L. and Figueiras-Vidal, A.R. (2010) Pattern classification with missing data: A review. *Neural Computing and Applications*, **19**, 263–282.

Geva, M., Schuster, R., Berant, J. and Levy, O. (2020) Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Gillioz, A., Casas, J., Mugellini, E. and Abou Khaled, O. (2020) Overview of the transformer-based models for NLP tasks, in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, IEEE Press, Piscataway, NJ, pp. 179–183.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2020) Generative adversarial networks. *Communications of the ACM*, **63**(11), 139–144.

Jiang, Y., Chang, S. and Wang, Z. (2021) Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, **34**, 14745–14758.

Kim, J., Tae, D. and Seok, J. (2020) A survey of missing data imputation using generative adversarial networks, in *2020 International conference on Artificial Intelligence in Information and Communication (ICAIIC)*, IEEE Press, Piscataway, NJ, pp. 454–456.

Lakshminarayan, K., Harp, S.A., Goldman, R. and Samad, T. (1996) Imputation of missing data using machine learning techniques. In *KDD-96 Proceedings*, AAAI, Palo Alto, CA, pp. 140–145.

Lee, D., Kim, J., Moon, W.-J. and Ye, J.C. (2019) CollaGAN: Collaborative GAN for missing image data imputation, in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ, pp. 2487–2496.

Li, Y., Dogan, A. and Liu, C. (2022) Ensemble generative adversarial imputation network with selective multi-generator (ESM-GAIN) for Missing data imputation, in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, IEEE Press, Piscataway, NJ, pp. 807–812.

Li, Y., Shi, Z., Liu, C., Tian, W., Kong, Z. and Williams, C.B. (2021) Augmented time regularized generative adversarial network (atr-gan) for data augmentation in online process anomaly detection. *IEEE Transactions on Automation Science and Engineering*, **19**(4), 3338–3355.

Liu, C., Kan, C. and Tian, W. (2020) An online side channel monitoring approach for cyber-physical attack detection of additive manufacturing, in *International Manufacturing Science and Engineering Conference*, American Society of Mechanical Engineers, Vol. 84263, p. V002T07A016.

Liu, C., Kong, Z., Babu, S., Joslin, C. and Ferguson, J. (2021) An integrated manifold learning approach for high-dimensional data feature extractions and its applications to online process monitoring of additive manufacturing. *IISE Transactions*, **53**(11), 1215–1230.

Lu, Q.Y. and Wong, C.H. (2018) Additive manufacturing process monitoring and control by non-destructive testing techniques: Challenges and in-process monitoring. *Virtual and Physical Prototyping*, **13**(2), 39–48.

Mazumder, R., Hastie, T. and Tibshirani, R. (2010) Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, **11**, 2287–2322.

Mirzaei, A., Carter, S.R., Patanwala, A.E. and Schneider, C.R. (2022) Missing data in surveys: Key concepts, approaches, and applications. *Research in Social and Administrative Pharmacy*, **18**(2), 2308–2316.

Musil, C.M., Warner, C.B., Yobas, P.K. and Jones, S.L. (2002) A comparison of imputation techniques for handling missing data. *Western Journal of Nursing Research*, **24**(7), 815–829.

Rao, P.K., 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 and Engineering*, **137**(6), 061007.

Sasaki, Y. and Fellow, R. (2007) *The truth of the f-measure, manchester: Mib-school of computer science*, University of Manchester, p. 25.

Shi, Z., Li, Y. and Liu, C. (2022) Knowledge distillation-enabled multi-stage incremental learning for online process monitoring in advanced manufacturing, in *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE Press, Piscataway, NJ, pp. 860–867.

Shi, Z., Mamun, A.A., Kan, C., Tian, W. and Liu, C. (2023) An LSTM-autoencoder based online side channel monitoring approach for cyber-physical attack detection in additive manufacturing. *Journal of Intelligent Manufacturing*, **34**, 1815–1831.

Stekhoven, D.J. and Bühlmann, P. (2012) MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, **28**(1), 112–118.

Sturm, L.D., Williams, C.B., Camelio, J.A., White, J. and Parker, R. (2017) Cyber-physical vulnerabilities in additive manufacturing systems: A case study attack on the. STL file with human subjects. *Journal of Manufacturing Systems*, **44**, 154–164.

Tay, Y., Dehghani, M., Bahri, D. and Metzler, D. (2022) Efficient transformers: A survey. *ACM Computing Surveys*, **55**(6), 1–28.

Vangipuram, R., Gunupudi, R.K., Puligadda, V.K. and Vinjamuri, J. (2020) A machine learning approach for imputation and anomaly detection in IoT environment. *Expert Systems*, **37**(5), e12556.

Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P.-A. (2008) Extracting and composing robust features with denoising autoencoders, in *Proceedings of the 25th International Conference on Machine Learning*, Association for Computing Machinery (ACM), New York, NY, pp. 1096–1103.

Yoon, J., Jordon, J. and Schaar, M. (2018) Gain: Missing data imputation using generative adversarial nets, in *International Conference on Machine Learning*, International Machine Learning Society (IMLS), Stockholm, Sweden, pp. 5689–5698.

Zhang, B., Gu, S., Zhang, B., Bao, J., Chen, D., Wen, F., Wang, Y. and Guo, B. (2022) Styleswin: Transformer-based gan for high-resolution image generation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, New Orleans, LA, pp. 11304–11314.

Zhang, H. and Woodruff, D.P. (2018) *Medical missing data imputation by Stackelberg gan*, Carnegie Mellon University.

Zhang, S. (2012) Nearest neighbor selection for iteratively kNN imputation. *Journal of Systems and Software*, **85**(11), 2541–2552.