Computers in Industry 153 (2023) 104024

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.sciencedirect.com/journal/computers-in-industry

Self-supervised representation learning anomaly detection methodology based on boosting algorithms enhanced by data augmentation using StyleGAN for manufacturing imbalanced data ☆

Yoonseok Kim [a,b], Taeheon Lee [a,b], Youngjoo Hyun [a,b], Eric Coatanea [c], Siren Mika [d], Jeonghoon Mo [b,1], YoungJun Yoo [a,*]

[a] Korea Inst. Industrial Tech. (KITECH), Republic of Korea
[b] Department of Information and Industrial Engineering, Yonsei University, Seoul, Republic of Korea
[c] Manufacturing Engineering and Systems Engineering, Tampere University, Finland
[d] VTT Technical Research Centre of Finland, Helsinki, Finland

ARTICLE INFO

Keywords:
Anomaly detection
Time-series data
Generative adversarial network
Boosting algorithm

ABSTRACT

This study proposes a methodology for detecting anomalies in the manufacturing industry using a self-supervised representation learning approach based on deep generative models. The challenge arises from the limited availability of data on defective products compared with normal data, leading to degradation in the performance of deep learning models owing to data imbalances. To address this limitation, we propose a process that leverages the Gramian angular field to transform time-series data into images, applies StyleGAN for image augmentation of anomalous data, and utilizes a boosting algorithm for classifier selection in supervised learning. Additionally, we compared the accuracy of the classifier before and after data augmentation. In experimental cases involving CNC milling machine data and wire arc additive manufacturing data, the proposed approach outperformed the approach before augmentation, resulting in improved precision, recall, and F1-score for anomaly detection. Furthermore, Bayesian optimization of the hyperparameters of the boosting algorithm further enhanced the performance metrics. The proposed process effectively addresses the data imbalance problem, and demonstrates its applicability to various manufacturing industries.

1. Introduction

Advances in computers and sensors have facilitated the handling of vast amounts of data, enabling reliable system management. Various applications are now collecting meaningful time-series data using sensors, with applications in weather forecasting, medicine, economics, and smart factories contributing to flexible system management (Choi et al., 2021). However, the analysis of time-series data from various sensors poses unique challenges.

Time-series data analysis must consider not only temporal dependencies but also sensor-to-sensor relationships. Traditional statistical and machine learning methods have successfully analyzed univariate time-series data by employing techniques such as seasonal and trend decomposition using loess (STL) (Cleveland et al., 1990), ARIMA (Box et al., 1970), and Holt–Winters exponential smoothing (Winters et al., 1960). However, rapid advancements in technology have resulted in a

☆ **Acknowledgements**

The study reported here was done within the Eureka! SMART project S0410 "Tools for adaptive and intelligent control of discrete manufacturing processes (TANDEM)", funded under the SMART Eureka Cluster on Advanced Manufacturing programme and supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government as "Tools for adaptive and intelligent control of discrete manufacturing process (TANDEM) (2/4)-P0022309". The Support is greatly acknowledged.

* Correspondence to: Korea Inst. Industrial Tech. (KITECH), 89, Yangdaegiro-gil, Ipjang-myeon, Seobuk-gu, Cheonan-si, Chungcheongnam-do, 31056, Republic of Korea.

E-mail addresses: j.mo@yonsei.ac.kr (J. Mo), youdalj@kitech.re.kr (Y. Yoo).

[1] **Postal address**: Engineering Building D Rm 1004, 50 Yonsei-Ro Seodaemun-Gu, Seoul, Korea 03722.

https://doi.org/10.1016/j.compind.2023.104024
Received 13 February 2023; Received in revised form 24 September 2023; Accepted 1 October 2023
Available online 8 October 2023

substantial increase in multivariate time-series data, rendering conventional analytical methods impractical. Deep learning, which is emerging as a solution to complex data problems, has witnessed remarkable achievements across various domains.

Recently, numerous studies have been conducted on time-series data analysis based on deep learning. The success of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) has resulted in an era of deep learning, and the variations of CNNs (Bai et al., 2018) and RNNs (Hochreiter et al., 1997) have achieved remarkable performance for classification and forecasting with sequence data that are difficult to analyze. This success has led to various applications of deep learning in time-series data analysis. However, deep learning models are highly dependent on the training data. In particular, time-series data analysis involves various problems with the training data. Moreover, collecting data over a long period inevitably results in an imbalance between the normal and abnormal data. In supervised learning, the imbalance between normal and anomalous data complicates model training. Therefore, unsupervised learning methods are widely used for time-series data analyses. LSTM-AE (Hsieh et al., 2019), MAD-GAN (Li et al., 2019), and LSTM-VAE (Park et al., 2018) based on RNN, BeatGAN (Zhou et al., 2019), TCN-GMM (Liu et al., 2019), and TCN-ms (He and Zhao, 2019) based on CNN, MSCRED (Zhang et al., 2019) and RSM-GAN (Khoshnevisan et al., 2020) based on ConvLSTM, and a mixture of two neural networks, MTAD-GAT (Zhao et al., 2020) based on GTA (Chen et al., 2021) and TFT (Lim et al., 2021) based on Attention are widely used for time-series data analysis.

In manufacturing processes, data scarcity and imbalance are especially prevalent challenges when applying deep learning models. Owing to the extended data collection periods, most of the available data pertain to normal operations, whereas only a limited portion corresponds to anomalies (Wen et al., 2021). This data imbalance poses obstacles to the development of reliable artificial intelligence models.

To address data imbalance, data augmentation methods are actively researched in the domain of deep learning. However, time-series data augmentation introduces additional complexity. Preserving time dependencies and accounting for intervariable relationships in multivariate time-series data augmentation are challenging. Furthermore, specific augmentation methods may be ineffective for anomaly detection in time-series data classification.

In real-world classification tasks, imbalanced data is a frequent challenge where the sizes of different classes' samples vary significantly. This issue is also encountered in the manufacturing industry, where data may have missing labels or suffer from class imbalances, leading to difficulties in the learning process. Researchers introduced self-supervised representation learning (SSRL) (Vincent et al., 2008; Pathak et al., 2016; Donahue et al., 2016), which comprises a pretext task and a downstream task to address data imbalance. In SSRL, the pretext task is initially solved, and then the model parameters obtained during this task are applied to the downstream task, which represents the ultimate problem that the user aims to solve. In the context of anomaly detection models that use the imbalance ratio (IR) problem, the pretext task involves the discovery of meaningful latent representations using generative models. The downstream task resulting from a pretext task is typically a classification model.

As an illustrative example from the domain of generative modeling, the pretext task entails reconstructing the original input while concurrently acquiring meaningful latent representations. The denoising autoencoder (Vincent et al., 2008) was purposefully designed to restore an image from a partially corrupted version or one that includes random noise. This design was inspired by the remarkable human capacity to recognize objects in pictures despite the presence of noise, suggesting the extraction and isolation of crucial visual features from noise. Another innovative approach is the context encoder (Pathak et al., 2016), which is trained to fill the missing regions of an image based on a binary mask. The model was trained using a combination of reconstruction and adversarial losses, with the mask defining the areas to be removed irrespective of their shape. The introduction of bidirectional generative adversarial networks (GANs) (Donahue et al., 2016) enhances the process by incorporating an additional encoder that learns the mapping from the input to the latent variable. This enhancement further refines the representation learning process, leading to an improved performance in downstream tasks.

GANs have demonstrated remarkable performance in various domains, including data generation, augmentation, and anomaly detection (Singh et al., 2020; Hertlein et al., 2021; Luo et al., 2021). Their success has been extended to the manufacturing industry, with applications such as defect data generation, design structure data generation, and mechanical defect data augmentation. In the biomedical sector, gathering data often requires deep expertise and a prolonged duration, making it challenging to combat data deficiencies. The gene expression generator (GEG) (Farou et al., 2020) is an innovative adaptation of the GAN, specifically tailored to generate data in the biomedical domain. By utilizing the GEG for data augmentation, noticeable improvements have been observed in the classification accuracy for various biomedical purposes. In the domain of speech emotion recognition (SER), the lack of diverse emotional speech data presents substantial performance constraints. An end-to-end text-to-speech (TTS) system built on an extended Tacotron 2 architecture (Latif et al., 2023) incorporated a conditional encoder in its generative model structure. This revamped TTS model showed a significant enhancement in the performance metrics for SER tasks. Originally conceptualized for computer vision tasks, GANs exhibit certain limitations when extrapolated to time-series data. Both discrete-variant and continuous-variant GANs (Brophy et al., 2023) have potential applications in an array of time-series datasets. FA-GAN (Li et al., 2023) introduced a pioneering framework capable of generating category texts. It strategically employs a feature encoder to extract the conditional contextual information, thereby facilitating the creation of diverse sentences. Simultaneously, the model utilizes a category encoder to embed categorical data, enabling control within the latent space, which underscores its potential for generating a multitude of category texts. GANs find utility in numerous domains and continually contribute to performance enhancement via a plethora of methodologies.

In this study, we propose an image data enhancement approach to augment time-series data utilizing the Gramian angular field (GAF) (Wang et al., 2015). This image data augmentation method transforms time-series data into images that are further augmented using a GAN. By balancing normal and anomalous data, this augmented time-series image data address data imbalance issues. We propose a machine learning-based framework for time-series anomaly detection using a balanced dataset. The proposed framework employs a SSRL approach that augments a limited amount of data, which are then utilized as training data for the anomaly detector. Among several boosting algorithms, the most accurate machine-learning model was selected for anomaly detection. Additionally, we compared the accuracy achieved with and without data augmentation by the GAN using boosting algorithms.

The remainder of this paper is structured as follows. Section 2 discusses previous research on data augmentation. Section 3 presents the proposed methods for the experimental data and the necessary steps. Section 4 details the experiments, validation methods, and results. Section 5 presents the experimental results. Finally, Section 6 provides the concluding remarks and outlines potential future research directions.

2. Related work

2.1. Augmentation of time-series data

Research on time-series data augmentation is being conducted continuously. Moreover, developing methods for augmenting various types of data while preserving the characteristics of the original data remains

a challenge. Until recently, the investigated time-series data augmentation methods could be broadly divided into two categories: basic and advanced approaches.

Basic approaches are methods that have been used in many fields for a long time and can be divided into three categories: time domain, frequency domain, and time–frequency domain. Basic approaches augment data in a rule-based manner, making it difficult to augment time-series data in the way that best represents their characteristics.

Advanced approaches can be divided into three broad categories: decomposition methods, statistical generative models, and learning-based methods. Decomposition methods typically employ STL or Robust STL (Wen et al., 2019). STL decomposes time-series data into trends, seasonality (periodic), and the remainder. The decomposed components are used to generate time-series data. Time-series data are generated using deterministic and stochastic components. The stochastic component is generated by building a residual-based statistical model, such as the autoregressive (AR) model.

2.2. Generative adversarial networks

GAN is a successful model in data generation modeling. In this method, adversarial learning is realized between a generator and discriminator, where the discriminator learns until it can no longer distinguish between real data and the fake data produced by the generator. In time-series data extensions, the basic learning method is the same, but the network structure of the GAN is changed.

Statistical generative models use statistical models to model time-series data. Most statistical generative models generate time-series data by assuming that the current timestamp is dependent on previous timestamps. In other words, it is difficult to expect good performance for multivariate or complex time-series data.

Learning-based models are time-series data augmentation models that utilize deep learning. Generative models include statistical models and neural networks. Currently, the neural networks primarily used for time-series data augmentation are enhanced with GANs. A GAN is a successful data generation model. The GAN trains in such a manner that the discriminator competes with the generator until it cannot distinguish between real data and the fake data produced by the generator. In time-series data augmentation, the GAN uses the same basic learning method and is examined in a manner that changes the network structure. A recurrent GAN (R-GAN) (ekri et al., 2019) for time-series data was examined to address the lack of time-series data and the defect problem. This was proposed as a way to set up a GAN in an LSTM to generate time-series data. A similar performance was shown for data trained on datasets generated by the R-GAN and models trained on real data, demonstrating the ability to generate GANs configured with recurrent neural networks. TimeGAN (Yoon et al., 2019) comprises four networks that generate time-series data. It consists of a sequence generator and discriminator using an RNN, embedding function, and recovery function. After obtaining the true latent code using the embedding function and the latent code generated by the sequence generator, the latent code is mapped to the time-series data using the recovery function. The sequence discriminator classifies the latent codes as true or false. By utilizing these four networks, the TimeGAN generates time-series data. In this study, we extend the data through a GAN that generates time-series images obtained via the GAF rather than a GAN using an RNN.

2.3. Boosting algorithms

Boosting is a prominent ensemble technique in machine learning that leverages the combination of multiple sequential weak learners to enhance prediction or classification performance. It aims to address the challenge of overfitting, in which a single model becomes excessively tailored to the training data. The fundamental principle of boosting ensembles is to aggregate the outcomes of weak models by training them on data and subsequently integrating their predictions or classifications for new instances.

Introducing a sequential aspect, boosting algorithms incorporate the idea of iteratively refining weak learners by explicitly considering the errors of preceding models. This concept is further expanded to gradient boosting machines (GBM), wherein weak learners are selected to continuously minimize the loss gradient.

Several boosting algorithms have been proposed, including XGBoost, adaptive boosting (AdaBoost), and gradient boost. Among them, XGBoost, which is based on gradient boosting, has gained significant recognition owing to its exceptional accuracy and efficiency. In the context of this study, one of the reasons for opting for XGBoost is its advanced optimization capability. It offers a fast and scalable tree-boosting system equipped with parallel processing, pruning, regularization, and missing-value handling. These characteristics are vital when dealing with high-dimensional and complex data, particularly in the anomaly detection domain. XGBoost has been successfully applied in diverse scientific domains to effectively address anomaly detection challenges.

AdaBoost focuses on training algorithms by assigning higher weig hts to misclassified instances, thus emphasizing on learning from challenging cases. Consequently, AdaBoost was specifically selected for its relevance to anomaly detection. It excels at classifying outliers that pose difficulties near the classification boundaries. Moreover, AdaBoost is widely acknowledged as a straightforward and efficient approach that enhances the prediction performance without introducing undue algorithmic complexity.

Gradient boosting, which is recognized as one of the most potent machine learning algorithms, constructs a robust ensemble model by combining multiple weak models. This is achieved by iteratively adjusting the predictions in a direction that minimizes the gradient of the loss function. This characteristic is particularly advantageous in the context of imbalanced data, which is a prevalent challenge in anomaly detection tasks. The superior performance, robustness, and adaptability to anomaly detection tasks contributed to the selection of gradient boosting in this study. Its effectiveness in maximizing the prediction accuracy and mitigating overfitting has been empirically established.

Within the proposed methodology, the aforementioned boosting algorithms are leveraged to augment unbalanced data using the latent space features generated by a GAN. These augmented features are then incorporated into a classifier, thereby improving the anomaly detection performance.

3. Methods

3.1. Problem statement

In this study, we considered anomalies in time interval w of multivariate time-series data with N variables. The time series data are written as $s = [s_t, \ldots, s_{t+w}]$. The time series data $s_t \in \mathbb{R}^N$ for time t are represented by the values of N variables from the N-dimensional vector. We converted s into a single image i. Our goal is to augment anomalous data to solve the data imbalance problem. StyleGAN (Karras et al., 2020) is used to learn the latent space of the converted image dataset I. We increase the anomaly detection performance by augmenting the anomalous data in the learned latent space. If there is an anomaly in the time interval, we label it as False (0).

3.2. Image embedding methodology for time-series data

As shown in Fig. 1, the framework of the proposed method is that time-series data are converted into images through the GAF, and the converted images are mapped to the latent space with a StyleGAN. Anomaly data were augmented in the latent space. Data augmentation solves data imbalance problems and creates an anomaly detector using boosting algorithms.

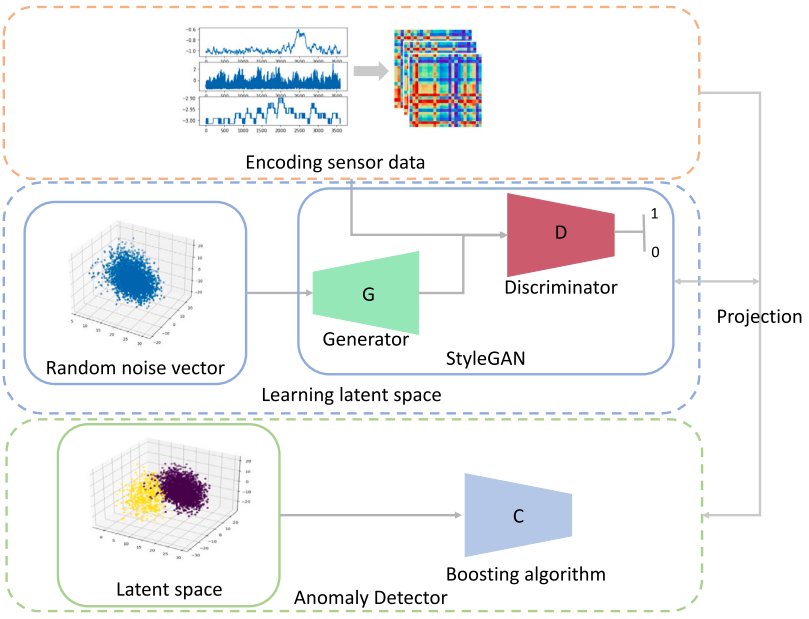

Fig. 1. Proposed framework for anomaly detection.

The method proposed in this study used the anomaly detection method after converting time-series data into image data. Recently, deep learning models that deal with image data have achieved remarkable progress compared with methods that deal with multivariate time-series data in the process of deep learning evolution. Therefore, the multivariate time-series data were converted into image data. The GAF method was used to convert the data into images.

The GAF expresses time-series values in terms of angles and radius, generates a Gramian matrix, and converts the matrix to an image by matching it to pixels. The time series data for the jth window for image transformation is represented by $S_j = [s_{t+1}, s_{t+2}, s_{t+3}, \dots, s_{t+w-1}, s_{t+w}]$, $s_t = [x_{1,t}, x_{2,t}, \dots, x_{N-1,t}, x_{N,t}]$, which is the vector of values for each sensor at time t, w denotes the observation window size, and N signifies the number of sensors. Each s_t is a scaled time series vector, denoted by $s_t = [x_{1,t}, x_{2,t}, \dots, x_{N,t}]$, which is transformed into an N-dimensional vector using an element-wise Arccos operation.

Time-series data are represented by $s_j \in \mathbb{R}^{N \times w}$ N variables and window size w. s_j is converted into one image i_j using the GAF, which then converts the time series into a matrix in two stages. The equations for these two stages are as follows:

$$\begin{cases} \phi = \arccos(s_t); \phi \in [0, \pi] \\ r = \frac{t}{w}; r \in \mathbb{R}^+, t \in w \end{cases} \tag{1}$$

First, the angle and radius of each time step are obtained, and then the Gramian matrix using the calculated angle and radius is calculated.

$$GAF = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \cdots & \cos(\phi_1 + \phi_N) \\ \vdots & \ddots & \vdots \\ \cos(\phi_N + \phi + 1) & \cdots & \cos(\phi_N + \phi_N) \end{bmatrix} \tag{2}$$

The elements of each matrix are computed as pixels and converted into an image, as shown in Fig. 2.

3.3. Generative adversarial network-based latent space learning

The time-series data converted into images are mapped to a low-dimensional representation space using a GAN. The learning process for the GAN follows this sequence. First, the collected normal and anomalous data are labeled and then fed into StyleGAN, which serves as a data representation tool. StyleGAN is used as a data representation tool because the latent space is a good representation of data features, and when learning with labels, the latent space is represented according to the labels. StyleGAN provides a representation of the latent vectors of input data. In the training process, convergence of the loss function is guaranteed even with a small amount of training data. The proposed study uses StyleGAN for the following reasons. The trained StyleGAN is used to represent the latent space. From the dataset collected using StyleGAN, the sample latent vectors z_{normal} and $z_{anomaly}$ in the latent space are extracted (sampled) according to the ratio of the dataset. The sampled latent vectors express the distributions of normal and abnormal situations in the latent space of the dataset, as shown in Fig. 3.

3.4. Data augmentation

The latent space of time-series data was obtained through the latent space learning process. The latent space obtained through the GAN was used to augment the data. Latent spaces are divided into anomalous and normal spaces. In time-series anomaly detection, the data imbalance between normal data and anomaly data affects the classification performance of machine learning algorithms. We extract data from the latent space to augment the data and solve data imbalance problems using the augmented data. Normal data with sufficient samples can be mapped to the latent vector $z_{normal} \in \mathbb{R}^{1 \times 512}$ using the StyleGAN. Using normal data n, a training dataset was created $X_{normal\_train} \in \mathbb{R}^{n \times 512}$.

The proposed methodology maps anomaly data to a latent vector $z_{anomaly} \in \mathbb{R}^{1 \times 512}$, just like normal data. $X_{anomaly\_train} \in \mathbb{R}^{m \times 512}$ should be configured using m anomalous data points ($m \ll n$) when anomalous data are much less than normal data. That is, in a given data space $X_{anomaly\_train}$, the data are augmented using the interpolation technique shown in Fig. 4. The training dataset $X_{train} \in \mathbb{R}^{2n \times 512}$ is obtained by matching the same dimensions of $X_{anomaly\_train}$ and X_{normal_train} to $n \times 512$ by augmenting anomaly data.

The data in the latent space z_{anomaly} is augmented using interpolation between the coordinates of each point. For example, as shown in Fig. 4, we obtained 512-dimensional latent vector values for the ith anomaly and the (i+1)th sample. If the ratio of normal to anomalous data is 3:1, which means an imbalance ratio of 3, the anomalous data should be doubled. To achieve this, we can set two interpolation points

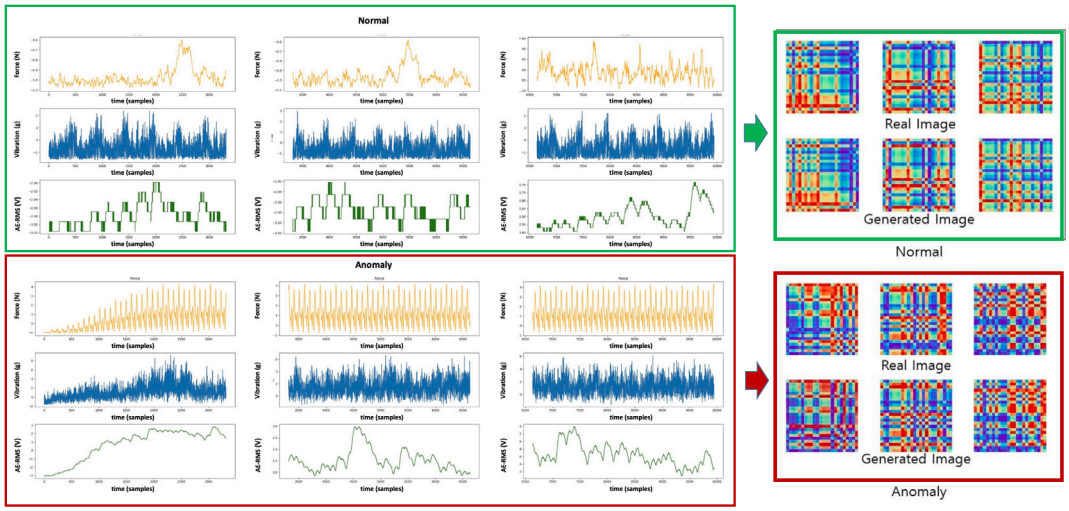

Fig. 2. Image augmented results of both the normal and the anomaly cases of the time series data using GAN.

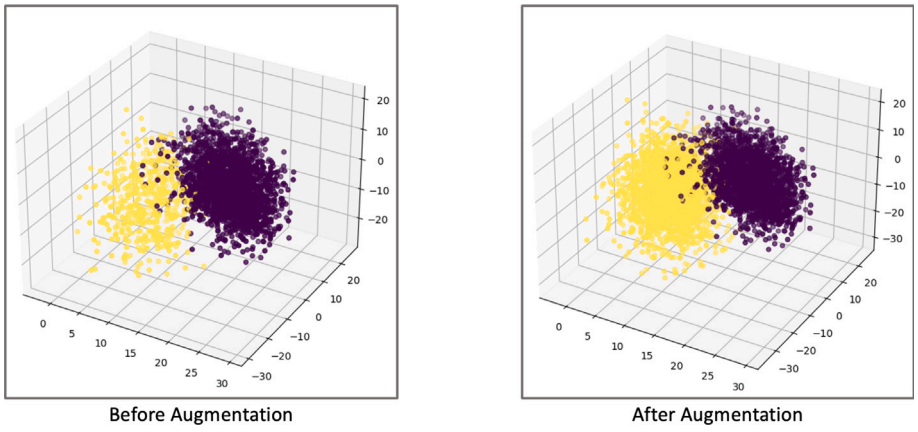

Fig. 3. Scatter plot of normal (purple) and anomaly (yellow) features before (left) and after (right) augmentation in latent space. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

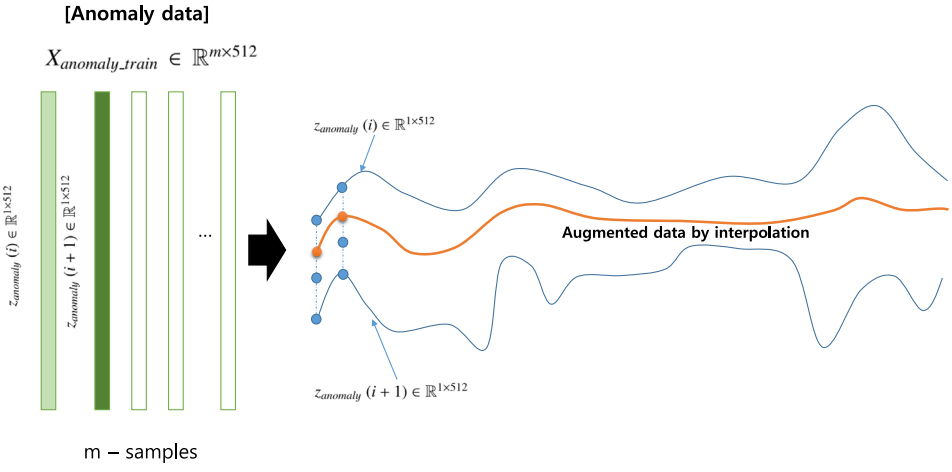

Fig. 4. Description of the data augmentation by interpolation of the anomaly data in latent space. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

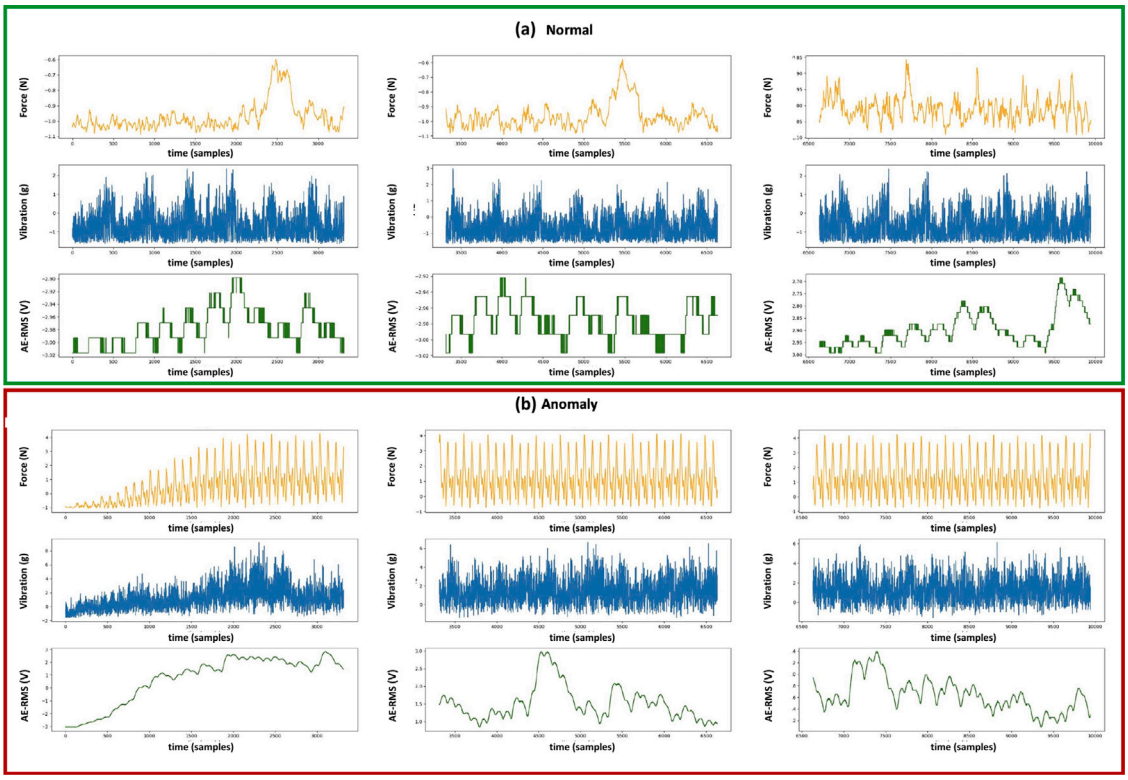

Fig. 5. Time-series data of the CNC milling machine: (a) normal cases, (b) abnormal cases.

between the *i*th anomaly and (i+1)th sample, as shown in Fig. 4, and generate augmented data (Fig. 4, orange line) using the interpolated latent vector values. This results in an equal ratio of normal-to-anomalous data. The quantity of augmented data can be controlled by adjusting the parameter that determines the number of interpolated latent vectors between the neighboring latent vectors in z_{anomaly}.

3.5. Anomaly detection algorithm

Training is performed by assigning the anomaly and the normal label to the training data set obtained from the latent space. By using data augmentation, we expect the dataset to be available for machine learning. We detected anomalies through classification using boosting algorithms. Data extracted from the latent space can solve this problem using a 512-dimensional feature data classification method. We employed the best boosting algorithm among various boosting algorithms as the classifier for our experimental technique.

The boosting (Chen et al., 2015) algorithm is one of the machine learning ensemble techniques that combine several weak learners and converts them into strong ones. The boosting algorithm learns the data based on the number of weak learners after random sampling with the first sample in the first learner and then uses the error with the second sample in the second learner to proceed with the learning. Overfitting is prevented as learning proceeds, and a strong learner can be obtained through iterative continuous learning.

In this study, we utilized three types of boosting algorithms: gradient boost, AdaBoost, and XGBoost, to classify augmented data in a reduced 512-dimensional latent vector space into anomaly and normal data. The gradient boost (Friedman, 2002) classifier uses decision trees as base learners. AdaBoost (Freund et al., 1996) is an adaptive boosting algorithm that assigns weights to poorly classified features using a gradient boost classifier. XGBoost (Chen et al., 2015), which was proposed to address the limitations of gradient boosting, is another algorithm considered in this study. After applying these three boosting algorithms, we selected the best boosting algorithm as the classifier.

4. Experimental results

The results of applying the proposed augmented-based anomaly detection technique to facility sensor data are detailed in the following sections.

4.1. Demonstration case 1: computer numerical control (CNC) milling machine data

4.1.1. Dataset and its pre-processing

In the paper, we utilized the PHM 2010 dataset (PHM Society, 2010) to validate the proposed method. We experimented involving milling machine sensor data $x_s = [f_x, f_y, f_z, v_x, v_y, v_z, AE] \in \mathbb{R}^{120000\times7}$, were f_x are the force values of the x-axis, y-axis, z-axis and, v_x, v_y, v_z represent force values of the x-axis, y-axis, z-axis and, AE is an acoustic emission sensor value, respectively. The sensor data from the milling machine were seven-dimensional, consisting of three-dimensional force, three-dimensional vibration, and AE-RMS. In this study, we converted the force and vibration data into vector size values. The ratio of normal data to anomalous data in the dataset used in the experiment was 5:1.

Fig. 5 illustrates the original time series of the CNC machining data. Fig. 5. a shows the normal cases and Fig. 5. b presents abnormal cases. When comparing abnormal and normal situations, it was apparent that the former exhibited higher values of vibration, force, and AE signals. In addition, the patterns exhibited greater fluctuations. These deviations can be attributed to increased tool wear on the CNC cutter (see Fig. 5).

For generating GAF images (Fig. 6), the input to the GAF consists of the 2-norm values of 7-dimensional vibrations, force, and acoustic emission, which are calculated as follows:

$$s_t = \sqrt{\|f_t\|^2 + \|v_t\|^2 + AE_t^2}, \tag{3}$$

$$\|f_t\| = \sqrt{f_x^2 + f_y^2 + f_z^2}, \tag{4}$$

$$\|v_t\| = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{5}$$

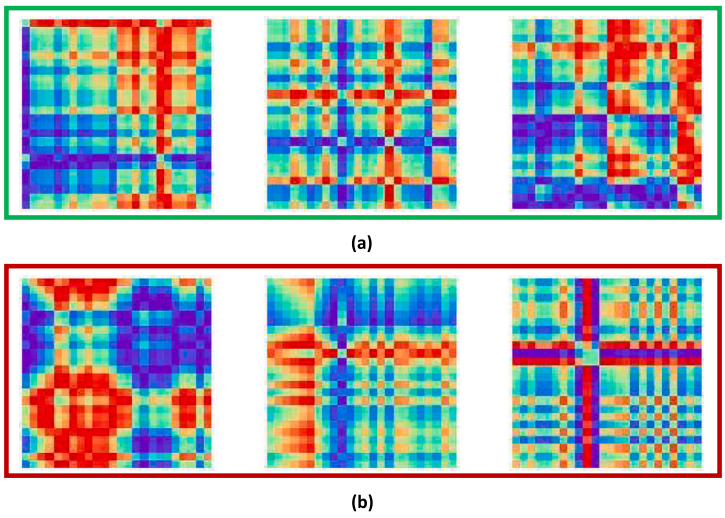

Fig. 6. GADF image of the CNC: (a) normal cases, (b) anormal cases.

The 2D representation of the GAF obtained using Eq. (2), which is the input of Eq. (3) in CNC machining, has a physical meaning that corresponds to the maximum energy of the signals.

When vibration, force, or sound increases, the input value of the GAF also increases. A large input value is then transformed into a 2D image of the GAF, which is displayed as a line-shaped value. By examining the characteristics of the signal, the 2D GAF representation allows for effective analysis in abnormal situations.

4.1.2. Evaluation and performance measures

We consider precision, recall, and F1-Score as classification performance evaluation metrics. The values were calculated using the following equation:

$$precision = \frac{TP}{TP + FP} \tag{6}$$

$$recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{8}$$

where TP, TN, FP, and FN are the numbers of true positives, true negatives, false positives, and false negatives, respectively. This is a good performance assessment method for the detection of data-unbalanced anomalies. It uses test data to evaluate accuracy.

4.1.3. Five-fold-cross-validation result for the experiments

In general, to create a machine learning model for improving its performance, we divide the dataset into training data (train set) and test data (test set). However, if we repeatedly evaluate the model's performance and adjust the parameters using a fixed test set, the model will eventually become one that performs well only on the test set. This phenomenon in which the model is overly trained on the test set and performs poorly on other real-world datasets is called overfitting.

To prevent such overfitting, cross-validation is necessary. By using cross-validation, we can make use of all the data in the dataset for training, thereby improving the model's performance and accuracy and creating a more generalized model.

The experiments were conducted using stratified 5-fold cross-validation. This approach distributes a dataset in a manner that maintains similar label distributions in the training and validation datasets. The performance of the model was evaluated using precision, recall, and F1-score as the evaluation/performance metrics for each fold. The results for the evaluation metrics are summarized in Table 1

The proposed methodology summarizes the results of showing the performance of the self-supervised representation learning classification by the two experiments. In the latent space of StyleGAN, before and after applying the proposed method (data augmentation), performance evaluations of anomaly detection and of each boosting algorithm were conducted. Table 1 compares the performance of the anomaly detector before and after the anomaly data augmentation. The ratio of data before augmentation was 5:1, indicating that the system lacked considerable anomalous data, as in a typical system.

When performing anomaly detector learning, if there is insufficient anomaly data, the value of recall, which represents the rate at which anomaly data was detected, is low as 0.9988 (XGBoost), 0.9848 (AdaBoost), and 0.9792 (gradient boost) in Table 1. If the anomaly detector determines that the anomaly is normal, it can be fatal to the system. Therefore, the performance of classifier models must be improved. After data augmentation to improve the performance, the evaluation metric (F1-score) of the anomaly detector improves significantly, and the recall increases for all models when the ratio of normal data to anomalous data is the same (1:1), as presented in Table 1. Thus, the data augmentation method may be a good alternative for analytical detection.

4.2. Demonstration case 2: WAAM welding data

For verification purposes, WAAM 3D printing data was utilized in Fig. 7. The data included the current, voltage, and wire feed rate (WFR) recorded across three channels. The normal and abnormal states can be identified by examining the 3D geometry of the printer. The material employed for the printing process was AM70, with a thickness of 1.2 mm. The target of the printing, the length of the printed wall, measures 160 mm. Geometric data, including 3D points and .stl files were exported from the laser scanner, specifically the HEXAGON METROLOGY 77 series. Fronius Explorer data consisted of event logs for each layer during the printing process. Logs were stored at a sampling rate of 0.1 s.

4.2.1. Dataset and preprocessing

The data consists of current, voltage, and wire feeding speed (WFS) data for normal WAAM (Fig. 7.a.1, a.2, and a.3). These data were used to validate the proposed process. These values are shown in (Fig. 7. a.1, a.2, and a.3). It is evident that the current, voltage, and WFS were maintained during normal WAAM. Specifically, when performing normal 3D printing, the current was 150 A, the voltage was 16 V,

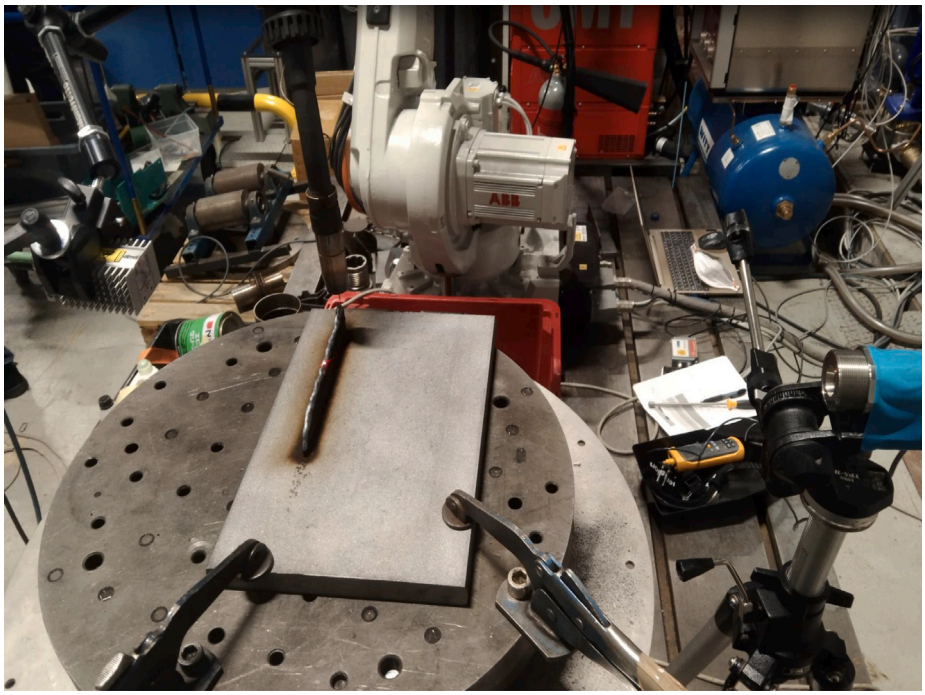

Fig. 7. Experimental setup for WAAM welding data.

Table 1
Anomaly detection accuracy in terms of precision (%), recall (%), and F1-score, on datasets with ground-truth anomalies for stratified 5-fold cross-validation for CNC data with imbalance ratio = 5.

Method		Before augmentation			After augmentation		
	Fold	Precision	Recall	F1-socre	Precision	Recall	F1-score
XGBoost	1	0.9967	0.9833	0.9899	0.9967	0.9967	0.9967
	2	0.9992	0.9958	0.9975	0.9992	0.9992	0.9992
	3	0.9933	0.9867	0.9899	0.9950	0.9950	0.9950
	4	0.9924	0.9825	0.9874	0.9975	0.9975	0.9975
	5	0.9992	0.9958	0.9975	0.9983	0.9983	0.9983
	Averaged	0.9962	0.9888	**0.9924**	0.9973	0.9973	**0.9973**
AdaBoost	1	0.9907	0.9742	0.9822	0.9909	0.9908	0.9908
	2	0.9959	0.9992	0.9975	0.9925	0.9925	0.9925
	3	0.9916	0.9783	0.9848	0.9901	0.9900	0.9900
	4	0.9891	0.9858	0.9875	0.9876	0.9875	0.9875
	5	0.9933	0.9867	0.9899	0.9942	0.9942	0.9942
	Averaged	0.9921	0.9848	**0.9884**	0.9911	0.9910	**0.9910**
Gradient Boost	1	0.9975	0.9875	0.9924	0.9967	0.9967	0.9967
	2	0.9959	0.9792	0.9873	0.9983	0.9983	0.9983
	3	0.9942	0.9708	0.9821	0.9926	0.9925	0.9925
	4	0.9942	0.9708	0.9821	0.9975	0.9975	0.9975
	5	0.9975	0.9875	0.9924	0.9975	0.9975	0.9975
	Averaged	0.9959	0.9792	**0.9873**	0.9965	0.9965	**0.9965**

and the WFS remained at approximately 5. However, in cases where the 3D printing result was defective, as shown in Fig. 8, there was an increase in the voltage and current values, and the WFS decreased to approximately 3.

For each layer, the time series results are shown for 26 layers in Fig. 9a.1, 30 layers in Fig. 9a.2, 40 layers in Fig. 9a.3, and 19 layers in Fig. 9b.1. Each layer comprised three channels and 114 data samples, resulting in a total measurement time of 11.4 s. Before converting the image, a window was set to truncate the data, ensuring a clear visibility of the feature point values (window size = 50). After normalization to a value between 0 and 1, GAF image embedding was performed. The image conversion results corresponding to the normal and abnormal voltage, current, and WFS situations are depicted in Fig. 10. The StyleGAN2-adaptive data augmentation (ADA) was trained based on these images.

4.2.2. Evaluation and performance measures

In this experiment, we also used the same performance evaluation metrics as in Section 4.1.2. We consider the precision (Eq. (6)), recall (Eq. (7)), and F1-score (Eq. (8)) as the classification performance evaluation metrics. We transformed the time series into GAF images using WAAM data and trained StyleGAN2-ADA with the transformed images. From StyleGAN2-ADA, we extracted latent vectors and extracted feature points. After data augmentation using StyleGAN2-ADA on the extracted latent vectors, we trained XGBoost, AdaBoost, and gradient boost.

4.2.3. Five-fold-cross-validation in the experiments

We performed the same process for WAAM data as outlined in Section 4.1.3. The experiments utilized a Stratified 5-fold cross-validation to ensure a comprehensive evaluation. Metrics (precision, recall, and

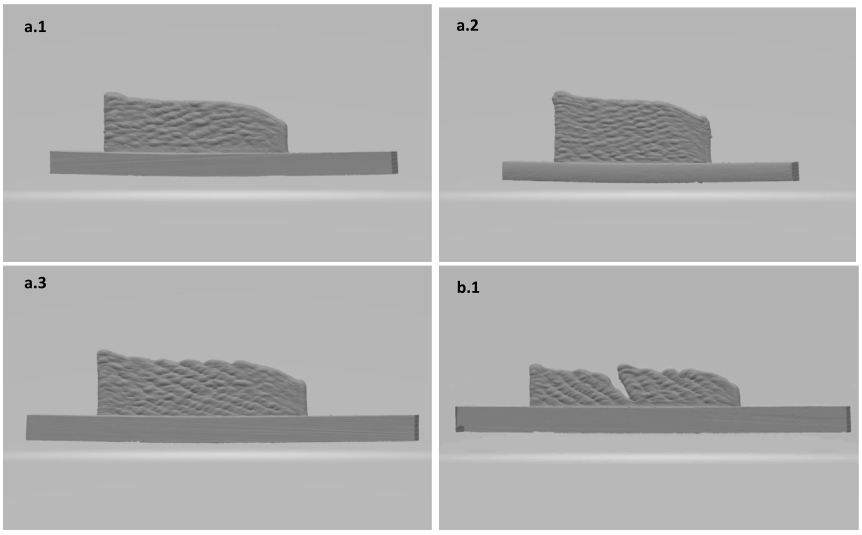

Fig. 8. Geometry of the WAAM experiment: normal cases (a.1, a.2, a.3) and anormal case (b.1).

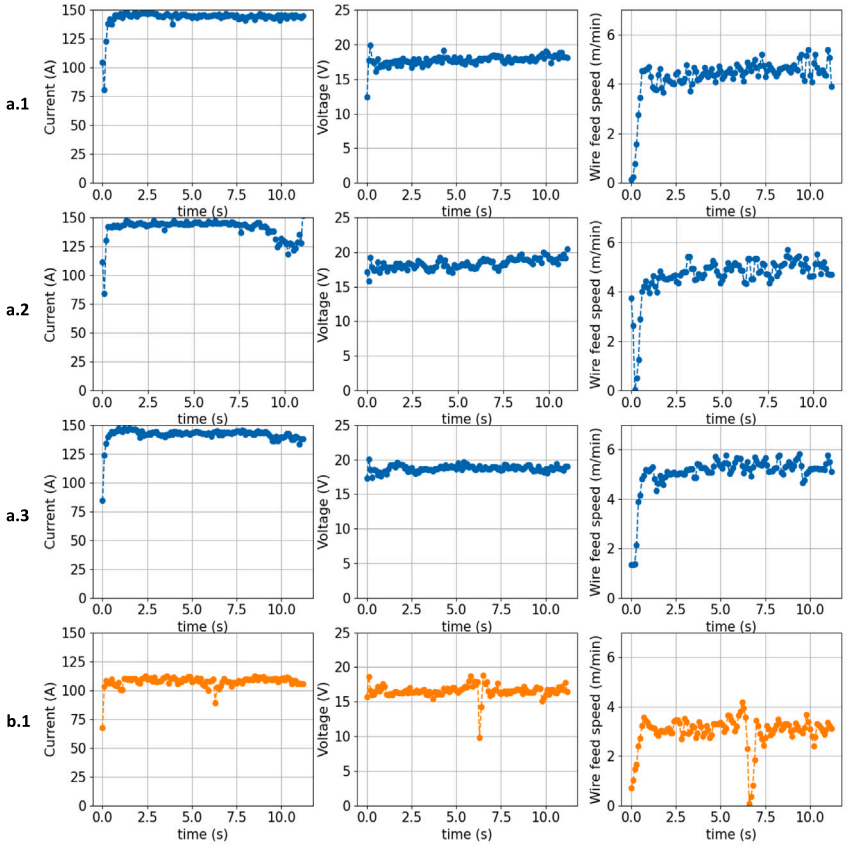

Fig. 9. Time-series of the WAAM: normal cases (a.1, a.2, a.3) and anormal case (b.1).

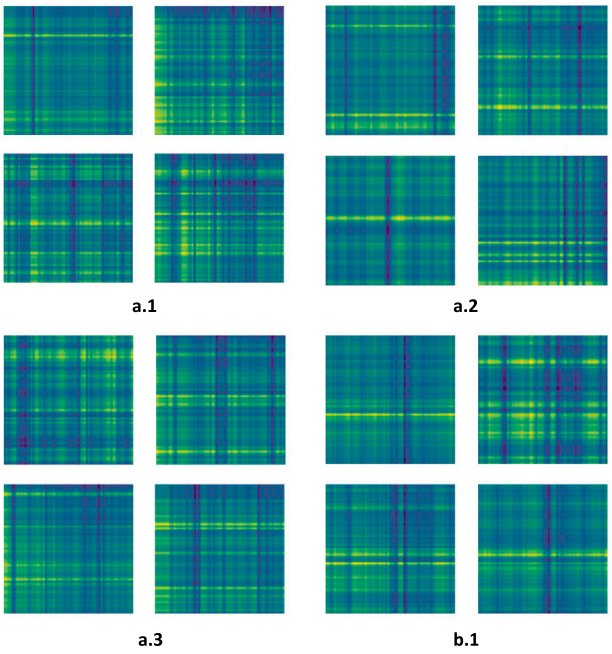

Fig. 10. GADF image of the WAAM: normal cases (a.1, a.2, a.3) and anormal case (b.1).

F1-score) were used to assess the performance of the model for each fold. The summarized results for these evaluation metrics can be found in Table 2.

The proposed methodology presents a comprehensive analysis of the self-supervised representation learning classification performance, accomplished through two experiments conducted in the latent space of StyleGAN. These experiments encompass the evaluation of the anomaly detection performance and effectiveness of each boosting algorithm, both before and after applying the proposed method, which incorporates data augmentation. Table 2 compares the performance of the anomaly detector before and after anomaly data augmentation. Before augmentation, the initial data ratio was 5:1, highlighting the significant scarcity of anomalous data, which is a common observation in similar systems.

Insufficient anomaly data during the training of the anomaly detector results in low recall values, indicating a poor rate of anomaly detection as demonstrated in Table 2 (0.8583 for XGBoost, 0.9667 for AdaBoost, and 0.7879 for gradient boost). After augmenting the data to enhance performance, there was a significant improvement in the evaluation metric (F1-score) of the anomaly detector. In addition, the recall increases for all models when the ratio of normal data to anomalous data is balanced at 1:1, as presented in Table 2. These results highlight the effectiveness of the proposed method.

5. Discussion about experiments results

5.1. Comparison with existing baseline limits

5.1.1. Image embedding of the GAF

Detecting anomalies by image embedding, like the Gramian angular field (GADF), offers advantages over analyzing time-series data. Image embedding provides a more intuitive interpretability, enabling domain experts to understand and track patterns easily. Anomalies that appear as single points in the time series can be represented as intersecting lines in the GADF, thus enhancing the visualization of anomalies. Spectrograms, such as short-time Fourier transforms (STFT), enable the depiction of the signal strength for repetitive cycles in an image format. Overall, leveraging image embedding and spectrograms improves the interpretability and representation of characteristic features in anomaly detection.

In the proposed approach, multivariate time series data is transformed into images using GAF, and anomaly detection is performed through GAN-based data augmentation. This method offers several key advantages over other techniques such as the Markov transition field (MTF) and short-time Fourier transform (STFT), which encode time series into images.

The first advantage of the proposed approach is the preservation of information. The GAF effectively preserves the temporal order and pattern information of the time-series data. While MTF is based on variations and STFT transforms time-series data into images using time-frequency analysis, these methods may sometimes fail to capture complex patterns or sequential information in the time domain. In contrast, the GAF effectively preserves such information, which is crucial for pattern-based anomaly detection.

The second advantage of the proposed approach is the ability to represent data visually. The GAF transformation allows visualization of the original time series data. This visual representation is important in the context of anomaly detection, because a GAN can easily process images. However, the visual representations produced by MTF and STFT may often be less intuitive.

The third advantage of the proposed approach is the ability to capture complex interactions. GAF effectively captures complex patterns and interactions in time-series data. This aspect is particularly emphasized during the transformation process into images, which are often missed by MTF and STFT.

Lastly, the proposed approach allows for efficient computation. The GAF is computationally efficient and enables the processing of large and complex time-series datasets. In contrast, STFT requires careful adjustment of the window size and overlap, whereas GAF does not have these tuning factors. Therefore, the advantages of GAF demonstrate why it is preferred over MTF or STFT in anomaly detection research.

Remark 1. The sensitivity of small anomalies varies depending on the resolution of the image embedding. With a 256×256 image size, the proposed image-embedding scheme can visualize local and small anomalies, particularly variations of 0.1 σ in the time series. To achieve this, we first normalized the time series during the data preprocessing stage, bringing it within the interval $[-1, 1]$. The normalized input is then embedded into GAF images.

5.1.2. Rationale for adopting StyleGAN2-ADA and hyperparameters used in verification

To facilitate a detailed analysis of local features, it is necessary to employ techniques such as Style-GAN and ProgressiveGAN, which excel at generating high-resolution images. StyleGAN excels in capturing intricate details. In contrast to traditional GANs, which rely on a single latent vector z as the input and cannot independently modify small specific parts, StyleGAN adopts a different architecture. It incorporates the concept of “w” obtained through the mapping network and inserts it into the style layer responsible for each resolution. This allows the expression of local feature points with high resolution.

Unlike traditional GANs that use a single latent vector “z” and cannot modify specific parts independently, StyleGAN employs a unique architecture. It integrates the latent vector of the mapping network into the style layer for each resolution, thereby allowing the precise expression of high-resolution local features. By contrast, it is difficult for the autoencoder (AE) to generate images that capture the temporal local features of the time series, as shown in Fig. 11.

In the proposed methodology, we utilized StyleGAN2-ADA for image generation in high-dimensional spaces. StyleGAN2-ADA is an enhanced version of StyleGAN2 that introduces methodologies for general GAN training. It addresses the issue of overfitting by adaptively adding noise to the training dataset and improving the image generation quality.

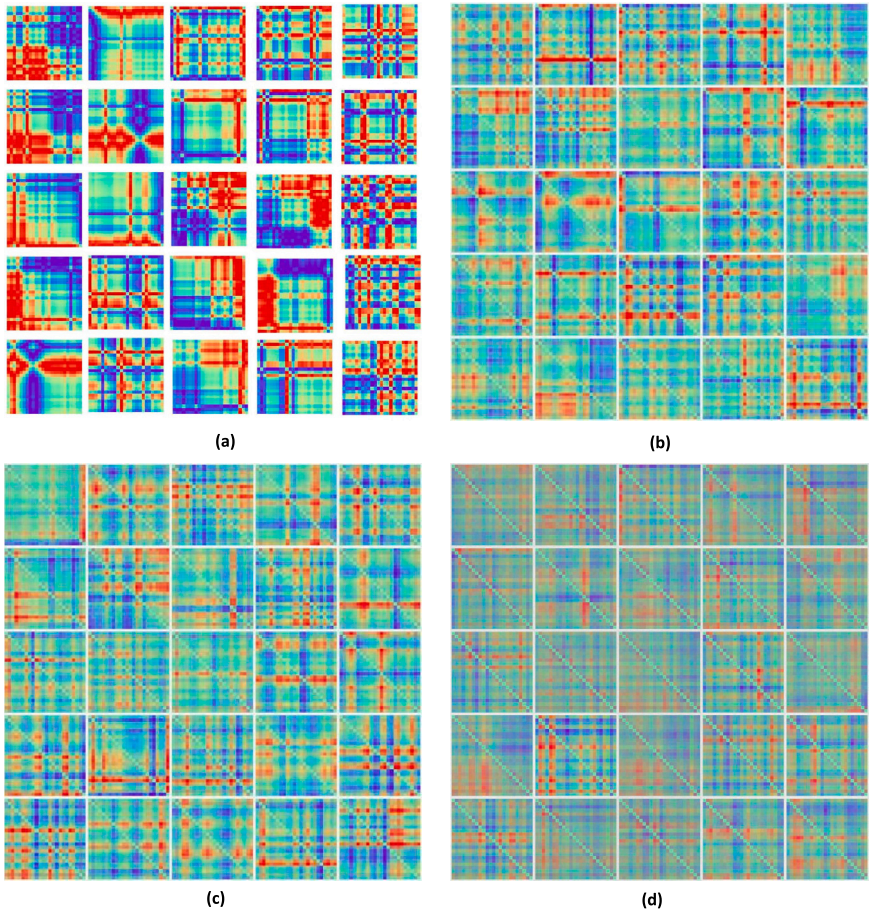

Fig. 11. Comparison of the image generating model: (a) proposed (StyleGAN2+ADA), (b) autoencoder, (c) beta-autoencoder, (d) variation autoencoder.

Table 2
Anomaly detection accuracy in terms of precision (%), recall (%), and F1-score, on datasets with ground-truth anomalies for stratified 5-fold cross-validation for WAAM data with Imbalance ratio = 5.

Method		Before augmentation			After augmentation		
	Fold	Precision	Recall	F1-socre	Precision	Recall	F1-score
XGBoost	1	0.9750	0.8750	0.9158	0.9979	0.9979	0.9979
	2	1.0000	1.0000	1.0000	0.9701	0.9686	0.9687
	3	0.9750	0.8333	0.8872	0.9979	0.9979	0.9979
	4	0.9750	0.8333	0.8872	0.9742	0.9728	0.9728
	5	0.9500	0.7500	0.807	1.0000	1.0000	1.0000
	Averaged	0.9750	0.8583	**0.8994**	0.9880	0.9874	**0.9875**
AdaBoost	1	1.0000	1.0000	1.0000	0.9959	0.9958	0.9958
	2	1.0000	1.0000	1.0000	0.9836	0.9833	0.9833
	3	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	4	0.975	0.8333	0.8872	0.9742	0.9728	0.9728
	5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	Averaged	0.9950	0.9667	**0.9774**	0.9907	0.9904	**0.9904**
Gradient Boost	1	0.7206	0.7961	0.7444	0.9959	0.9958	0.9958
	2	0.8333	0.9474	0.8722	1.0000	1.0000	1.0000
	3	0.9524	0.6667	0.7250	1.0000	1.0000	1.0000
	4	0.8070	0.8070	0.8070	0.9742	0.9728	0.9728
	5	0.7807	0.7222	0.7452	0.9979	0.9979	0.9979
	Averaged	0.8188	0.7879	**0.7788**	0.9936	0.9933	**0.9933**

Table 3
Anomaly detection accuracy using SMOTE algorithm with CNN (ResNet50V2 backbone) in terms of precision (%), recall (%), and F1-score, on datasets with ground-truth anomalies for Stratified 5-fold cross-validation for CNC data with Imbance ratio = 5.

Method		Before augmentation			After augmentation		
	Fold	Precision	Recall	F1-socre	Precision	Recall	F1-score
XGBoost	1	0.2667	0.0667	0.1067	0.4494	0.1667	0.2432
	2	0.3103	0.0792	0.1262	0.2787	0.0708	0.1130
	3	0.3443	0.0875	0.1395	0.2262	0.0792	0.1173
	4	0.3333	0.0917	0.1438	0.3099	0.0917	0.1415
	5	0.3636	0.0833	0.1356	0.2923	0.0792	0.1246
	Averaged	0.3236	0.0817	**0.1304**	0.3113	0.0975	**0.1479**
AdaBoost	1	0.3582	0.1000	0.1564	0.3265	0.0667	0.1107
	2	0.3065	0.0792	0.1258	0.2885	0.0625	0.1027
	3	0.2812	0.0750	0.1184	0.2321	0.0542	0.0878
	4	0.3699	0.1125	0.1725	0.3333	0.0750	0.1224
	5	0.4375	0.1167	0.1842	0.2034	0.0500	0.0803
	Averaged	0.3507	0.0967	**0.1515**	0.2768	0.0617	**0.1008**
Gradient Boost	1	0.2000	0.0125	0.0235	0.3077	0.0167	0.0316
	2	0.3103	0.0375	0.0669	0.1667	0.0125	0.0233
	3	0.2308	0.0125	0.0237	0.3750	0.0250	0.0469
	4	0.3478	0.0333	0.0608	0.2105	0.0167	0.0309
	5	0.4074	0.0458	0.0824	0.2727	0.0125	0.0239
	Averaged	0.2993	0.0283	**0.0515**	0.2665	0.0167	**0.0313**

Table 4
Anomaly detection accuracy in terms of precision (%), recall (%), and F1-score, on datasets with ground-truth anomalies for stratified 5-fold cross-validation for IR ratio of CNC data.

Method		XGBoost		
	Fold	Precision	Recall	F1-socre
IR = 10.0	1	0.9958	0.9958	0.9958
	2	0.9901	0.9900	0.9900
	3	0.9950	0.9950	0.9950
	4	0.9918	0.9917	0.9917
	5	0.9975	0.9975	0.9975
	Averaged	0.9940	0.9940	**0.9940**
IR = 5.0	1	0.9967	0.9967	0.9967
	2	0.9992	0.9992	0.9992
	3	0.9950	0.9950	0.9950
	4	0.9975	0.9975	0.9975
	5	0.9983	0.9983	0.9983
	Averaged	0.9973	0.9973	**0.9973**
IR = 2.5	1	0.9935	0.9933	0.9934
	2	1.0000	1.0000	1.0000
	3	0.9983	0.9979	0.9981
	4	0.9979	0.9983	0.9981
	5	0.9973	0.9971	0.9972
	Averaged	0.9974	0.9973	**0.9974**

Table 5
Anomaly detection accuracy in terms of precision (%), recall (%), and F1-score, on datasets with ground-truth anomalies for stratified 5-fold cross-validation for IR ratio of WAAM data.

Method		Gradient Boost		
	Fold	Precision	Recall	F1-socre
IR = 2.5	1	0.9979	0.9974	0.9977
	2	0.9800	0.9738	0.9764
	3	0.9839	0.9791	0.9811
	4	1.0000	1.0000	1.0000
	5	0.9979	0.9740	0.9764
	Averaged	0.9883	0.9849	**0.9863**
IR = 5.0	1	0.9959	0.9958	0.9958
	2	1.0000	1.0000	1.0000
	3	1.0000	1.0000	1.0000
	4	0.9742	0.9728	0.9728
	5	0.9979	0.9979	0.9979
	Averaged	0.9936	0.9933	**0.9933**
IR = 10.0	1	0.9898	0.9958	0.9958
	2	0.9959	0.9958	0.9958
	3	1.0000	1.0000	1.0000
	4	0.9686	0.9664	0.9664
	5	1.0000	1.0000	1.0000
	Averaged	0.9909	0.9933	**0.9903**

One distinguishing feature of StyleGAN2-ADA lies in its architecture, which comprises a style-based generator and a multi-step discriminator. This architectural configuration enables the style-based generator to exert precise control over distinct styles in each layer, affording fine-grained manipulation of the synthesized images. Moreover, through the implementation of adversarial learning dynamics between the generator and discriminator, StyleGAN2-ADA facilitates the production of high-quality images, further enhancing its effectiveness in image synthesis.

During the training process, we employed the R1 regularization term to optimize the implemented network. This regularization method alleviates mode collapse issues and improves the quality of the generated images.

In our experimental setup, we adhere to the default configuration of StyleGAN2-ADA. The image resolution was set to 256 × 256 pixels, striking a balance between visual fidelity and computational efficiency. With a batch size of 16, we ensured efficient optimization using the Adam optimizer with a learning rate of 0.001 to update the network parameters effectively. These choices were aligned to achieve optimal performance and quality in the synthesis of high-dimensional images.

5.1.3. Comparison with image classification performance of CNN model

In this subsection, we present a comparison of two distinct methodologies aimed at evaluating the effectiveness of augmentation techniques within the latent space. The first methodology is the proposed method, which is a detection approach that uses the data augmentation of latent vectors using StyleGAN. The second methodology focuses on anomaly detection by augmenting time-series data using the synthetic minority oversampling technique (SMOTE).

Table 6
Hyperprameter of the XGBoost model.

Hyperparameter	Default value	Range	Descrpition
colsample_bytree	1	[3, 10]	*colsample_bytree* is the subsample ratio of columns when constructing each tree. Subsampling occurs once for every tree constructed.
gamma	0.0	[0,5]	A node is split only when the resulting split gives a positive reduction in the loss function.
max_depth	6.0	[3, 10]	The maximum depth of a tree.
min_child_weight	1	[1, 10]	It defines the minimum sum of weights of all observations required in a child.
reg_alpha	0.0	[0,1]	L1 regularization term on weights analogous to Lasso regression.

Table 7
Hyperparameter of the gradient boost model.

Hyperparameter	Default value	Range	Description
min_samples_split	2	[2, 10]	minimum number of samples required to split an internal node.
min_samples_leaf	1	[1,10]	Minimum number of samples required to be at a leaf node.
min_weight_fraction_leaf	0.0	[0, 0.5]	Minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node.
max_depth	3	[1, 10]	Maximum depth of the individual regression estimators.
min_impurity_decrease	0.0	[0.0, 3.0]	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.

Table 8
Hyperparameter of the XGBoost model for CNC data.

Hyperparameter	Default value	Optimized value
colsample_bytree	1	1
gamma .	0.0	0.07736
max_depth	6.0	3.0
min_child_weight	1	1.0
reg_alpha	0.0	1.0

For the augmentation of time series data, we employed the SMOTE technique to enrich anomalous time series datasets. Subsequently, the augmented time-series data were encoded using the advanced GAF method.

The encoded data was then employed as input for a ResNet50V2 backbone model, a deep neural network architecture that finds widespread application in contemporary research. Binary classification training is performed by leveraging this backbone architecture. Throughout the training process, the backbone model extracted feature vectors with 512 dimensions, which were subsequently harnessed to construct an anomaly detector using the XGBoost algorithm. The decision to configure the feature vector dimension to 512 was motivated by the desire to align it with the dimensionality of the latent vectors in StyleGAN.

As a result, it was observed that the anomaly detector based on time series data augmentation exhibited relatively lower performance (Table 3) compared to the latent space augmentation-based anomaly detector (Table 1). This phenomenon can be interpreted as arising from the complexity of augmenting time-series data, considering the interactions between sensors and temporal dynamics inherent to time-series data.

5.2. Guideline of the imbalance ratio

In this paper, for the CNC data, the first verification was conducted using a dataset with an IR of 5 for the normal and abnormal states. For the WAAM data, the verification was performed using a dataset with an IR of 5. This is a common scenario in manufacturing where the majority of the data represent the normal state.

It was observed in Zhu et al. (2020) that as the IR value increases, the classification result deteriorates. To address this, the proposed approach generates data with an IR of 1 by creating image embeddings using Style-GAN2-ADA. The results of the experiment on the IR value can serve as a guide for other researchers applying this approach in practice.

5.2.1. Demonstration case 1: computer numerical control (CNC) milling machine data

Table 4 presents the performance of XGBoost when trained on CNC data using 5-fold cross-validation. This demonstrates that as the difference between the distributions of normal and anomalous data decreases (indicated by a lower IR value that approaches 1), there is an improvement in the precision, recall, and F1-score performance. For instance, in terms of the averaged F1-score, we observed values of 0.9974 (IR = 2.5), 0.9973 (IR = 5.0), and 0.9940 (IR = 10.0). This indicates that the best performance was achieved when the IR value was 2.5.

5.2.2. Demonstration case 2: WAAM data

Similarly to the findings in the previous subsection, Table 5 presents cases of the gradient boost performance when trained on WAAM data using 5-fold cross-validation. As presented in Table 5, upon applying the gradient boost, it is apparent that the F1-score increases when the imbalance ratio (IR) is set to 5.0, compared to IR = 10.0. However, at IR = 2.5, the F1-score experiences a decline to 0.9863. This outcome can be explained by the influence of the data size. The CNC data (Table 4) underwent experimentation with a dataset comprising 120,000 samples, and the WAAM (Table 5) featured a small dataset with only 94 samples for normal data and 20 samples for anomalies for each layer.

Table 9
Anomaly detection accuracy in terms of precision (%), recall (%), and F1-score, on datasets with ground-truth anomalies for stratified 5-fold cross-validation for CNC data with Imbalance ratio = 5.

Method		Before hyperparameter optimization			After hyperparameter optimization		
	Fold	Precision	Recall	F1-socre	Precision	Recall	F1-score
XGBoost	1	0.9967	0.9967	0.9967	0.9992	0.9992	0.9992
	2	0.9992	0.9992	0.9992	0.9992	0.9992	0.9992
	3	0.9950	0.9950	0.9950	0.9942	0.9942	0.9942
	4	0.9975	0.9975	0.9975	0.9992	0.9992	0.9992
	5	0.9983	0.9983	0.9983	0.9983	0.9983	0.9983
	Averaged	0.9973	0.9973	**0.9973**	0.9980	0.9980	**0.9980**

Table 10
Hyperparameter of the gradient boost model for WAAM data.

Hyperparameter	Default value	Optimized value
min_samples_split	2	7
min_samples_leaf	1	9
min_weight_fraction_leaf	0.0	0.2137
max_depth	3	10
min_impurity_decrease	0.0	1.4879

To address this, the WAAM dataset was subject to augmentation via StyleGAN2-ADA, resulting in 1200 normal data samples and 120, 240, and 480 anomaly data samples for IR = 10, IR = 5, and IR = 2.5, respectively. These augmented data were then leveraged to extract latent vectors, upon which a classifier was trained to utilize boosting algorithms.

In scenarios where significant data augmentation is performed on a limited dataset, there exists a potential risk. Even with augmentation using StyleGAN2-ADA, the generated images may not accurately encapsulate defective data. If the augmented images predominantly represent defective instances, this could lead to a decline in classifier performance. Hence, when augmenting defective data using StyleGAN2-ADA for a small dataset, it is imperative to determine an appropriate augmentation ratio to uphold classifier efficacy.

5.3. Hyperparameter tuning

We proceeded with hyperparameter tuning. Among the boosting algorithms, AdaBoost, gradient boost, and XGBoost, we found that XGBoost had the highest F-1 score, which served as a performance metric. Consequently, we utilized Bayesian optimization to fine-tune the hyperparameters of XGBoost for the CNC data and gradient boost for WAAM data.

Bayesian optimization is an approach employed to identify the optimal set of parameters for machine learning or deep learning algorithms. It explores the hyperparameter space to determine the values that minimize the loss function. Five parameters were subject to optimization, and Tables 6 and 7 provide detailed explanations, default values, and specified ranges for optimization purposes.

5.3.1. Demonstration case 1: computer numerical control milling machine data

We optimized five hyperparameters of XGBoost on the CNC dataset using Bayesian optimization. Table 8 lists the default and optimized values of the five hyperparameters. Table 9 summarizes the performance metrics obtained through 5-fold evaluation. From Table 9, it is evident that utilizing the optimized hyperparameters leads to an increase in the F1-score from 0.9973 to 0.9980, compared to the default values. This confirms the efficacy of the proposed hyperparameter optimization in improving the performance metrics and enhancing the reliability of the classification model.

5.3.2. Demonstration case 2: WAAM data

Similar to the previous subsection, we optimized five hyperparameters of gradient boost on the WAAM dataset using Bayesian optimization. In Table 10, we present the default values along with the optimized values for these five hyperparameters. Table 11 provides an overview of the performance metrics acquired via the 5-fold evaluation. As presented in Table 11, there is an improvement in the F1-score, which increases from 0.9933 to 0.9983 with the utilization of the optimized hyperparameters, in contrast to the default values.

6. Conclusion

In this study, we propose a novel method for detecting anomalies using data augmentation techniques and manifold inverse mapping in the latent space. The key feature of the proposed approach is its ability to address the data imbalance problem by augmenting the data in the latent space with both normal and abnormal data, which are effectively trained using StyleGAN.

By employing the proposed method with the data imbalance problem resolved, we demonstrate its successful anomaly detection capability on the CNC and WAAM datasets. The results showed a promising performance in identifying anomalies in unbalanced datasets.

However, to further validate the effectiveness and applicability of our method to a wider range of scenarios, we plan to conduct additional research focusing on high-dimensional data. This allowed us to thoroughly assess the performance and potential limitations of the method in more complex and diverse datasets.

The significance of this study lies in its innovative approach to anomaly detection, which not only addresses the data imbalance challenge but also leverages latent space augmentation for improved accuracy and generalization. With the success achieved in this study, we believe that the proposed method has significant potential for practical applications in various real-world scenarios involving anomaly detection and classification tasks.

Funding

This study was performed using Eureka! SMART project S0410 “Tools for adaptive and intelligent control of discrete manufacturing processes (TANDEM)”, funded under the SMART Eureka Cluster on Advanced Manufacturing programme and supported by a Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government as “Tools for adaptive and intelligent control of discrete manufacturing process (TANDEM) (2/4)- P0022309”. Their Support is gratefully acknowledged.

Declaration of competing interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article

Data availability

The data that has been used is confidential.

Table 11
Anomaly detection accuracy in terms of precision (%), recall (%), and F1-score, on datasets with ground-truth anomalies for stratified 5-fold cross-validation for WAAM data with Imbalance ratio = 5.

Method		Before hyperparameter optimization			After hyperparameter optimization		
	Fold	Precision	Recall	F1-socre	Precision	Recall	F1-score
Gradient Boost	1	0.9959	0.9958	0.9958	0.9959	0.9958	0.9958
	2	1.0000	1.0000	1.0000	0.9979	0.9979	0.9979
	3	1.0000	1.0000	1.0000	0.9979	0.9979	0.9979
	4	0.9742	0.9728	0.9728	1.0000	1.0000	1.0000
	5	0.9979	0.9979	0.9979	1.0000	1.0000	1.0000
	Averaged	0.9936	0.9933	**0.9933**	0.9983	0.9983	**0.9983**

References

Bai, S., et al., 2018. Empirical evaluation of generic convolutional and recurrent networks for sequence modeling. http://dx.doi.org/10.48550/arXiv.1803.01271, arXiv preprint arXiv:1803.01271.

Box, G.E., et al., 1970. Distribution of residual autocorrelations in autoregression-integrated moving average time-series models. J. Amer. Statist. Assoc. 65 (332), 1509–1526. http://dx.doi.org/10.1080/01621459.1970.10481180.

Brophy, E., et al., 2023. Generative adversarial networks in time series: A systematic literature review. ACM Comput. Surv. 55 (10), 1–31. http://dx.doi.org/10.1145/3559540.

Chen, T., et al., 2015. XgBoost: Extreme gradient boosting. pp. 1–4, R package version 0.4-2. 1(4).

Chen, Z., et al., 2021. Learning graph structures with transformers for multivariate time-series anomaly detection in IoT. IEEE Internet Things http://dx.doi.org/10.1109/jiot.2021.3100509.

Choi, K., et al., 2021. Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. IEEE Access http://dx.doi.org/10.1109/access.2021.3107975.

Cleveland, R.B., et al., 1990. STL: Seasonal trend decomposition. J. Off. Stat. 6 (1), 3–73.

Donahue, J., et al., 2016. Adversarial feature learning. In: Int. Conf. on Learn. Representations. http://dx.doi.org/10.48550/arXiv.1605.09782.

ekri, M.N., et al., 2019. Generation of energy data for machine learning with recurrent generative adversarial networks. Energy 13 (1), 130. http://dx.doi.org/10.3390/en13010130.

Farou, Z., Mhoub, N., Horváth, T., 2020. Data were generated using a gene expression generator. In: Intelligent Data Engineering and Automated Learning–IDEAL 2020: 21st International Conference, Guimaraes, Portugal, Nov. 4–6, Proceedings, Part II 21. Springer Int. Publ., pp. 54–65. http://dx.doi.org/10.1007/978-3-030-62365-4_6.

Freund, Y., et al., 1996. Experiments using the new boosting algorithm. In: ICML, Vol. 96. pp. 148–156.

Friedman, J.H., 2002. Stochastic gradient boosting. Comput. Statist. Data Anal. 38 (4), 367–378. http://dx.doi.org/10.1016/S0167-9473(01)00065-2.

He, Y., Zhao, J., 2019. Temporal convolutional networks for anomaly detection in time series. J. Phys.: Conf. Ser. 1213 (4), 042050. http://dx.doi.org/10.1088/1742-6596/1213/4/042050.

Hertlein, N., et al., 2021. Generative adversarial network for early stage design flexibility in topology optimization for additive manufacturing. J. Manuf. Syst. 59, 675–685. http://dx.doi.org/10.1016/j.jmsy.2021.04.007.

Hochreiter, S., et al., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Hsieh, R.J., et al., 2019. Unsupervised online anomaly detection using multivariate sensing time-series data for smart manufacturing. In: IEEE 12th Conf. on Service-Oriented Computers. Appl. (SOCA). pp. 90–97. http://dx.doi.org/10.1109/soca.2019.00021.

Karras, T., et al., 2020. Training Generative Adversarial Networks with limited data. Adv. Neural Inf. Process. Syst. 33, 12104–12114. http://dx.doi.org/10.48550/arXiv.2006.06676.

Khoshnevisan, F., et al., 2020. Improving the robustness of seasonality-heavy multivariate time series anomaly detection. http://dx.doi.org/10.48550/arXiv.2007.14254, arXiv preprint arXiv:2007.14254.

Latif, S., et al., 2023. Generative emotional AI for speech emotion recognition: The case of synthetic emotional speech augmentation. Appl. Acoust. 210, 109425. http://dx.doi.org/10.1016/j.apacoust.2023.109425.

Li, D., et al., 2019. MAD-GAN: Multivariate anomaly detection for time-series data using networks. In: Int. Conf. Artif. Neural Netw.. pp. 703–716. http://dx.doi.org/10.1007/978-3-030-30490-4_56.

Li, X., et al., 2023. Feature-aware conditional GAN for category-text generation. Neurocomputing 547, 126352. http://dx.doi.org/10.1016/j.neucom.2023.126352.

Lim, B., et al., 2021. Temporal Fusion Transformers for interpretable multi-horizon time-series forecasting. Int. J. Forecast. 37 (4), 1748–1764. http://dx.doi.org/10.1016/j.ijforecast.2021.03.012.

Liu, J., et al., 2019. Anomaly detection for time series using temporal convolutional networks and Gaussian mixture model. J. Phys.: Conf. Ser. 1187 (4), 042111. http://dx.doi.org/10.1088/1742-6596/1187/4/042111.

Luo, J., et al., 2021. Case study of conditional deep convolutional generative adversarial networks for machine fault diagnosis. J. Intell. Manuf. 32 (2), 407–425. http://dx.doi.org/10.1007/s10845-020-01579-w.

Park, D., et al., 2018. Multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. IEEE Robot. Autom. Lett. 3 (3), 1544–1551. http://dx.doi.org/10.1109/lra.2018.2801475.

Pathak, D., et al., 2016. Context encoders: Feature learning by inpainting. In: Proc. of the IEEE Conf. on Comput. Visual and Pattern Recognition. pp. 2536–2544. http://dx.doi.org/10.1109/cvpr.2016.278.

[Dataset] PHM Society, 2010. PHM data challenge 2010.

Singh, R., et al., 2020. Generative Adversarial Networks for synthetic defect generation in assembly and test manufacturing. In: 2020 31st Annual SEMI Advanced Semiconductors Manuf. Conf. (ASMC). pp. 1–5. http://dx.doi.org/10.1109/asmc49169.2020.9185242.

Vincent, P., et al., 2008. Extraction and composition of robust features using Autoencoders. In: ICML '08: Proc. of the 25th Int. Conf. for Machine Learning. pp. 1096–1103. http://dx.doi.org/10.1145/1390156.1390294.

Wang, Z., et al., 2015. Time series are encoded as images for visual inspection and classification using tiled convolutional neural networks. In: Workshops at the Twenty-Ninth AAAI Conf. on Artif. Intell..

Wen, Q., et al., 2019. RobustSTL: A robust seasonal trend decomposition algorithm for long time series. In: Proc. of the AAAI Conf. on Artif. Intell., Vol. 33. pp. 5409–5416. http://dx.doi.org/10.1609/aaai.v33i01.33015409, (01).

Wen, Q., et al., 2021. Time-series data augmentation for deep learning: A survey. In: Proc. of the Thirtieth Int. Joint Conf. on Artif. Intell. Survey Track. pp. 4653–4660. http://dx.doi.org/10.24963/ijcai.2021/631.

Winters, P.R., et al., 1960. Forecasting sales using exponentially weighted moving average. Manage. Sci. 6 (3), 324–342. http://dx.doi.org/10.1287/mnsc.6.3.324.

Yoon, J., et al., 2019. Time-series Generative Adversarial Networks. Adv. Neural Inf. Process. Syst. 32, http://dx.doi.org/10.5555/3454287.3454781.

Zhang, C., et al., 2019. Deep neural networks for unsupervised anomaly detection and diagnosis in multivariate time-series data. In: Proc. AAAI Conf. Artif. Intell., Vol. 33. pp. 1409–1416. http://dx.doi.org/10.1609/aaai.v33i01.33011409, (01).

Zhao, H., et al., 2020. Multivariate time-series anomaly detection using a graph attention network. In: IEEE Int. Conf. Data Min. (ICDM). pp. 841–850. http://dx.doi.org/10.1109/icdm50108.2020.00093.

Zhou, B., et al., 2019. BeatGAN: Anomalous rhythm detection using adversarially generated time series. In: IJCAI. pp. 4433–4439. http://dx.doi.org/10.24963/ijcai.2019/616.

Zhu, R., et al., 2020. The imbalance ratio is adjusted using the dimensionality of the imbalanced data. Pattern Recognit. Lett. 133, 217–223. http://dx.doi.org/10.1016/j.patrec.2020.03.004.