An Interpretable Pipeline for Imbalanced Industrial Anomaly Detection: VAE-GAN Augmentation, CatBoost Classification, and TreeSHAP Interpretation

By

Dong-Jun, Chen

MSc Robotics Dissertation

School of Engineering Mathematics and Technology
UNIVERSITY OF BRISTOL
&
School of Engineering
UNIVERSITY OF THE WEST OF ENGLAND

A MSc dissertation submitted to the University of Bristol and the University of the West of England in accordance with the requirements of the degree of MASTER OF SCIENCE IN ROBOTICS in the Faculty of Engineering.

August 29, 2025

Declaration of own work

I declare that the work in this MSc dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Dong-Jun, Chen August 29, 2025

Ethics statement

This project did not require ethical review, as determined by my supervisor Carwyn Ward

Dong-Jun, Chen August 29, 2025

An Interpretable Pipeline for Imbalanced Industrial Anomaly Detection: VAE-GAN Augmentation, CatBoost Classification, and TreeSHAP Interpretation

Dong-Jun, Chen

Abstract— **In advanced smart manufacturing, anomaly detection is of paramount importance for ensuring product quality. However, class imbalance caused by the rarity of anomalous events, together with challenges from unknown anomalies, hinders the generalization of anomaly detection models. Previous studies also lack interpretabilty in their detection results. Hence, this study proposes a novel interpretable end-to-end anomaly detection pipeline, integrating with Variational Autoencoder-Generative Adversarial Network (VAE-GAN) for data augmentation, Categorical Boosting (CatBoost) for classification, and TreeSHAP for explainable AI. Leveraging the UCI-SECOM semiconductor dataset, the pipeline first performs preprocessing through exploratory data analysis (EDA), reducing feature dimensionality by 94.4%. Subsequently, VAE-GAN is deployed to generate anomalous samples, with quality surpassing vanilla VAE baselines, quantified using four distance metrics. CatBoost is then applied with Bayesian hyperparameter optimization, achieving 98.5% precision, 95.3% recall, and 96.9% F1-score, outperforming model using vanilla VAE and prior SMOTE-generated tree-based models. TreeSHAP provides visualization of global feature importance and instance-level anomalous attributions. Empirical validation confirms the model's deployment feasibility through online simulated streaming data. Although VAE-GAN may be limited by mode collapse issue, future direction could explore diffusion model as alternatives. This framework advances interpretable anomaly detection, offering significant implications for robust data-driven decision-making in smart manufacturing.**

I. Introduction

The Industrial Internet of Things (IIoT) integrates sensors, communication technologies, and cloud analytics to digitize manufacturing, enabling data-driven decision-making across various sectors. However, reliance on smart manufacturing introduces risks. Minor defects in production process, such as semiconductor wafer issues, cause significant financial losses[1], while undetected anomalies lead to unplanned downtime, costing industries roughly $50 billion annually [2]. Thus, anomaly detection is critical to ensure product quality and mitigate risks in smart manufacturing.

Anomaly detection methods are divided into model-based and data-driven approaches [3]. Model-based methods rely on domain experts' knowledge, but are time-consuming and lack generalization. Data-driven methods, leveraging machine learning (ML), include supervised and unsupervised learning approaches and require no prior device knowledge. However, their performance is limited by scarce anomalous data in real-world industrial settings.

To address class imbalance, data augmentation techniques have been deployed to generate diverse samples from minority anomalous data using transformation-based or generative-based approaches [4]. Transformation-based methods are simple to implement, while generative-based methods produce more realistic samples, but may be unstable. A study indicates generative AI (GenAI) shows instability in an industrial signal detection, missing 95% of data patterns, thereby exacerbating class imbalance and reducing detection performance [5].

Anomaly detection systems can effectively identify anomalies but often lack interpretability, making it challenging for domain experts to trust and adopt these models, particularly in high-risk domains [6]. A case study demonstrated that satellite operators were reluctant to entrust decisions involving millions of dollars to anomaly detection models lacking transparency [7]. To enhance domain experts' trust in these models, Explainable AI (XAI) has become a focus in this study.

Building on the current research, this study proposes a novel end-to-end anomaly detection pipeline to address class imbalance with high accuracy and interpretability. Specifically, it integrates VAE and VAE-GAN for realistic data generation, CatBoost for anomaly detection, and TreeSHAP (Shapley Additive Explanation) for explainable insights. This pipeline is evaluated on ”UCI-SECOM” semiconductor dataset [8], reflecting real industrial scenarios. Prior studies often used SMOTE with various detection model on this dataset [9]. Hence, this study employs VAE-GAN as an improved data augmentation method, compared against VAE as the baseline and prior SMOTE-based studies, to enhance performance. Thus, The aim of this study is:

1) To identify critical features in high-dimensional sensors data to reduce the computational cost of model training and inference.
2) To verify that VAE-GAN captures real anomalous distributions better than a vanilla VAE.
3) To quantify how synthetic data quality influences anomaly detection performance.
4) To determine the optimal oversampling ratio for the best detection performance.
5) To provide global and local explanations.
6) To validate the model's online detection capability.

To achieve the above research aim, their corresponding quantitative method and experiment objectives is:

1) To perform EDA technique to reveal feature correlation and class distribution gaps.

2) To compare VAE and VAE-GAN generated quality by four divergence metrics : Kullback-Leibler (KL) divergence, Jensen-Shannon (JS) divergence, Wasserstein distance, and maximum mean discrepancy (MMD).
3) To evaluate CatBoost under VAE and VAE-GAN data augmentation using Precision, Recall, F1-score, and other classification metrics.
4) To analyze and quantify varying oversampling ratios (0%, 30%, 50%, 80%, 100%) on CatBoost performance.
5) To apply TreeSHAP to visualize global feature importances and instance-level explanations of identified anomalies, presented via bar/beeswarm and waterfall plots.
6) To simulate industrial deployment by streaming raw normal and anomalous data to the model and interpreting detected anomalies.

II. Literature Review

A. Data Augmentation

Data augmentation enhances sample diversity, reduce overfitting, and improves classification in class-imbalanced scenarios. Traditional methods (e.g., rotation, flipping, scaling) generate new samples but struggle with high-dimensional data [10]. SMOTE improved minority attack detection, achieving an F1-score of 0.824, but lacks realism in high-dimensional settings [11]. In contrast, generative methods generate realistic samples. S. Fan et al. [12] proposed using VAE to generate defective-wafer traces, outperforming SMOTE-generated samples. GAN is utilized to generate realistic anomalous images in additive manufacturing, enhancing supervised anomaly detection performance [13]. However, VAE suffers a blurring issue, resulting in overly smooth samples, while GAN encounters mode collapse, limiting samples diversity. This study adopts VAE-GAN, combining VAE and GAN strengths to generate high-quality samples while mitigating their limitations.

B. Anomaly Detection Model

Various models have been developed for precise detection. Traditional model-based methods, like Statistical Control Charts (SCCs), shorten the average detection time and improve detection sensitivity, without increasing the false alarm rate [14]. A Principal Component Analysis (PCA)-based method quantifies deviation with low complexity in IoT environments [15]. Although these methods are computationally simple and interpretable, they fail to handle nonlinear data due to linear assumptions.

In contrast, data-driven methods leverage labeled data to train classification models, achieving superior performance with high-quality labels. A Gaussian Process (GP) model attains 100% accuracy with confidence bands, but at high computational cost [16]. Random Forest achieves the highest F1-score and lowest modeling cost among KNN and SVM, though it is less interpretable [17]. Gradient Boosting Decision Trees (GBDTs)-based methods outperform generic ML in manufacturing settings [18]. However, these methods rely on precise labels and are prone to target leakage, limiting generalization.

Unsupervised methods detect anomalies without labels by assuming distinct distributions between normal and anomalous data [19]. A pioneering study by M. Jeon et al. [20] combined Autoencoder (AE) with Dynamic Time Warping (DTW), augmented by digital twin data, for time-series anomalies. CNN-VAE model [21] enables real-time detection of spatiotemporal anomalies in industrial robot. LSTM predicts sensor reading, identifying anomalies from residuals [22]. However, these methods struggle with overlapping distributions and high computational costs, hindering deployment in resource-constrained environments.

This study adopts CatBoost for anomaly detection, leveraging its build-in algorithms to mitigate target leakage and overcome limitations of unsupervised methods reliant on distinct data distributions.

C. Explainable AI (XAI)

Valid interpretations enhance model truthworthiness. Previous study have used LIME [23] to interpret cybersecurity anomalies in autonomous vehicles with low computational cost and high local completeness, but it lacks global accuracy and stability. SHAP, applied to AE, visualizes feature contributions with high reconstruction error, offering great robustness than LIME despite higher computational cost [24]. Thus, this study employs TreeSHAP, leveraging its tree-based architecture to effectively compute SHAP value, addressing the computational cost issue.

III. Background

A. VAE

VAE, proposed by D. P. Kingma et al. [25], is employed as a data augmentation method to generate synthetic anomalous samples and mitigate class imbalance. A VAE consists of an encoder and a decoder. The encoder compresses each high dimensional input x into a lower dimensional latent vector z. The decoder then maps the latent vector z back to an output close to the input data x.

The objective of a VAE is to approximate the data distribution by maximizing the evidence lower bound (ELBO). In practice, we minimize the negative ELBO, which serves as the VAE loss. Hence, the loss function is given in Equation(1).

$$\begin{aligned}\mathcal{L}_{\text{VAE}} &= -\int q(z \mid x)\,\log\!\left(\frac{P(x \mid z)P(z)}{q(z \mid x)}\right)\,dz \\ &= \mathcal{L}_{\text{recon}} \;+\; \mathcal{L}_{\text{prior}}\end{aligned} \tag{1}$$

Here, $\mathcal{L}_{recon}$ is the reconstruction loss (Equation(2)). $\mathcal{L}_{prior}$ is the regularization term (Equation(3)), keeping $q_{\phi}(z \mid x)$ close to the prior $p(z)$.

$$\mathcal{L}_{\text{recon}} = -\mathbb{E}_{q(z|x)}\big[\log P(x \mid z)\big] \tag{2}$$

$$\mathcal{L}_{\text{prior}} = D_{\text{KL}}(q(z \mid x) \,\|\, P(z)) \tag{3}$$

However, VAE models each pixel with an independent Gaussian or Bernoulli likelihood, treating every output dimension as independent. To minimize the loss, the model often chooses the mean value, making the image lose detailed structure and thus causing blurring. For tabular data in this study, the generated samples lack sharp anomaly features needed for realistic anomalies, since extreme or rare patterns are averaged out.

B. VAE-GAN

VAE-GAN, proposed by A. B. L. Larsen et al. [26], serves as an effective data augmentation approach by combining VAE's inference ability with GAN's high-fidelity generation. Specifically, GANs, proposed by I. J. Goodfellow et al. [27], suffer from mode collapse, reducing diversity in the generated distribution. Additionally, the absence of an encoder limits explicit inference and reconstruction error calculation. VAE-GAN overcome these issues by integrating the structures of VAE and GAN with three components: Encoder, Decoder/Generator, and Discriminator.

1) **Encoder** (E): Maps input data x into a latent space z, enforcing the KL regularizer to keep the variational posterior close to the standard-normal prior $p(z) = \mathcal{N}(0, I)$, as shown in Equation(3).
2) **Decoder/Generator** (DG): Combines VAE's decoder and GAN's generator, reconstructing x as $\tilde{x} = DG(\tilde{z})$ during inference and generating new samples from prior noise $\hat{x} = DG(z)$. Moreover, it minimizes reconstruction loss using a feature-space loss from an intermediate discriminator layer D_{layer} to alleviate blurring from pixel-wise MSE in VAEs, as shown in Equation(4).

$$\mathcal{L}_{DG_{recon}} = -\mathbb{E}_{q(z|x)}\big[\log P(D_{layer}(x) \mid z)\big] \tag{4}$$

3) **Discriminator** (D): Similar to a vanilla GAN, it outputs a scalar to distinguish real samples from generated ones, and also receives reconstructed $\tilde{x}$ and treats it as fake. Consequently, the adversarial loss is modified as in Equation(5).

$$\begin{aligned} \mathcal{L}_{GAN} = \log D(x) & \\ + \log(1 - D(DG(\tilde{z}))) & \\ + \log(1 - D(DG(z))) & \end{aligned} \tag{5}$$

In summary, VAE-GAN is trained through a minimax game to produce diverse samples with fine detail.

C. CatBoost

In anomaly detection, traditional ML suffers from target leakage and high cardinality categorical features, hindering model's generalization capability. CatBoost, proposed by L. Prokhorenkova et al. [28] effectively addresses these problems.

Traditional gradient boosting algorithms use target values of all training samples to compute gradients, leading to target leakage. Specifically, the prediction distribution for a training sample $\mathbf{x_k}$ is influenced by its target value y_k, resulting in a discrepancy with the prediction distribution for test samples $\mathbf{x}$. To tackle this issue, CatBoost introduces Ordered Boosting, a variant that employs a random permutation σ to order the training samples, computing gradients only from preceding samples to eliminate target leakage.

For high cardinality categorical features (e.g., device types) in industrial sites, traditional one-hot encoding or gradient statistics methods struggle with these features, leading to dimensionality explosion. CatBoost addresses this challenge with Ordered Target Statistics (Ordered TS), converting categorical features into numerical representations. It employs multiple random permutations $(\sigma_1, \sigma_2, \ldots, \sigma_s)$ across iterations to reduce variance in target statistics and enhance stability. This ensures consistent statistics across training and test data while utilizing all training data effectively.

D. TreeSHAP

S. M. Lundberg et al. proposed SHAP [29], offering feature-level attributions using Shapley value from cooperative game theory. It quantifies the expected model output increase when adding each feature.

However, SHAP values requires enumerating all 2^M feature subsets, causing a burden of high computational complexity of $\mathcal{O}(2^M)$. To alleviate this cost, TreeSHAP [30], designed for tree ensembles, utilizes split structures of trees to recursively evaluate the conditional expectations $\mathbb{E}[f(x) \mid x_s]$ without enumerating every subset, as illustrated in Equation(6), reducing computational complexity to $\mathcal{O}(TLD^2)$, where T is the number of trees, L is the number of leaves, and D is the tree depth. Additionally, TreeSHAP provides interaction values, enabling visualization of pairwise feature interactions.

$$\mathbb{E}\big[f(x) \mid x_S\big] = \\ \begin{cases} w\,r_j, & v_j \neq \text{internal}, \\ G(a_j, w), & d_j \in S,\ x_{d_j} \leq t_j, \\ G(b_j, w), & d_j \in S,\ x_{d_j} > t_j, \\ G(a_j, w r_{a_j}/r_j) + G(b_j, w r_{b_j}/r_j), & d_j \notin S. \end{cases} \tag{6}$$

Here, j denotes a tree node, with w as the path weight and r_j as the leaf value.

IV. Implementation

A. System Overview

In real industrial scenarios, rare and subtle anomaly samples within vast normal data are often overlooked, leading to severe outcomes. To address the degraded detection performance due to class imbalance in device datasets, this study proposes an end-to-end pipeline, shown in Fig.(1), using UCI-SECOM datasets. In the offline training phase, raw data is preprocessed, and real anomaly samples are extracted for augmentation with VAE and VAE-GAN to generate synthetic anomalous samples, mitigating class imbalance. CatBoost is then trained as anomaly detection model to compare the performance of both augmentation methods, with resulting model preserved for online validation. TreeSHAP analyzes the output to identify key features and store the corresponding explainer. After model training, the system simulates real-time inference via an API. Input raw data undergoes the same preprocessing procedure before being evaluated by CatBoost to determine anomalies. If detected, TreeSHAP generates a waterfall plot to intuitively present the key anomaly feature points and their contribution to domain experts.

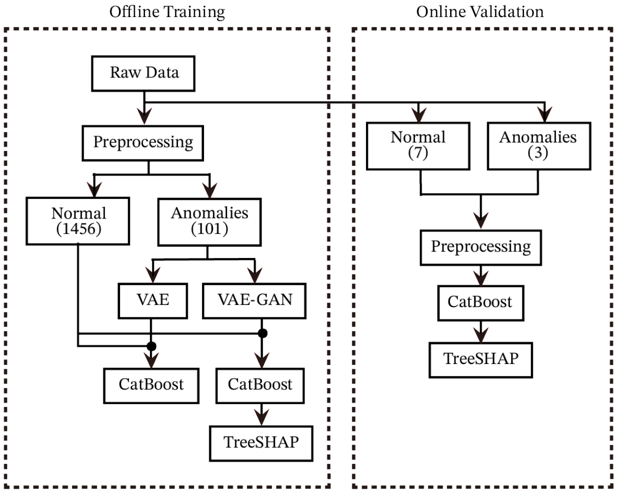

Fig. 1. System Flow Chart

B. Data Preprocessing

Before conducting data preprocessing, 10 raw data samples, consisting of 7 normal and 3 anomalous, are extracted to preserve realism in real industrial scenarios for online validation. First, features with a missing rate above 20% are removed to avoid over-reliance on imputed values. Subsequently, all numerical features are imputed with the mean values to address the missing value problem. Next, to mitigate the high computational burden from high dimensionality, this study sequentially (i) removes features with variance below 0.05, (ii) excludes features with an absolute Pearson correlation coefficient with the label below 0.05, (iii) retains only the most informative feature among pairs with high correlation ($|r| > 0.75$). Lastly, all retained features are standardized using z-score ($\mu = 0, \sigma = 1$). After these steps, the feature dimensionality is reduced from 590 to 33 (a 94.4% reduction), and 6% real anomalous samples are extracted for subsequent VAE and VAE-GAN based minority anomaly generation.

C. Anomalous Samples Generation

To compare the generative capability of VAE and VAE-GAN on tabular data, this study deliberately designs the same architectures for both models to eliminate structure interference.

For the VAE architecture, the encoder uses a fully connected network with node counts decreasing from 512 to 32 layer by layer, incorporating batch normalization and L2 regularization between layers to mitigate over-fitting and stabilize convergence. The decoder completely mirrors the encoder, with the final layer having no activation function to ensure the reconstructed output remains continuous and matches the original features dimensionality. The VAE loss function, shown in Equation (7), introduces a β coefficient to control the trade-off between latent space regularization and reconstruction. VAE hyperparameters are chosen from commonly used ranges, as detailed in Table(I).

$$\mathcal{L}_{VAE} = \beta \times \mathcal{L}_{prior} + \mathcal{L}_{recon} \tag{7}$$

The VAE-GAN further incorporates a 5-layer discriminator with the same depth as the encoder and outputs a normal/anomaly probability. Additionally, according to the concept in Section III-B, feature vectors are extracted from the discriminator's second hidden layer for feature-matching loss to enhance the quality of reconstructed samples. The loss function of VAE-GAN is divided into three parts: the encoder loss $\mathcal{L}_{Enc}$, generator loss $\mathcal{L}_{Gen}$, and discriminator loss $\mathcal{L}_{Dis}$, as shown in Equation(8), (9), and (10), respectively. Furthermore, γ is introduced to trade off between reconstruction and adversarial loss. All three are designed to minimize the loss so that the generated anomalous samples closely approximate real anomalous samples.

$$\mathcal{L}_{Enc} = \mathcal{L}_{prior} + \mathcal{L}_{DG_{recon}} \tag{8}$$

$$\mathcal{L}_{Gen} = \gamma \times \mathcal{L}_{DG_{recon}} - \mathcal{L}_{GAN} \tag{9}$$

$$\mathcal{L}_{Dis} = \mathcal{L}_{GAN} \tag{10}$$

For hyperparameter selection, the hyperparameters of VAE-GAN were aligned with those of VAE, as derailed in Table(I), to ensured a controlled comparison.

D. CatBoost Anomaly Detector

When the issue of class imbalance is addressed, CatBoost is employed as the core technique, outlined in Section(III-C). The dataset, comprising original and generated anomaly data, is divided into 80%/20% training and testing sets, respectively. Bayesian optimization is then employed to identify the best hyperparameter set, defining a six hyperparameters searching space: (i) max depth $\in$ [1, 5] (ii) iterations $\in$ [1, 3000] (iii) learning rate $\in$ [0.001, 0.5] (iv) L2 regularization $\in$ [3, 10] (v) min data in leaf $\in$ [1, 20] (vi) scale pos weight $\in$ [1, 50] Therefore, the objective function $f(\theta)$, where θ = (max_depth, iterations, learning_rate, l2_leaf_reg, min_data_in_leaf) aims to maximize the F1-score via 5-fold cross-validation, formulated as in Equation(11).

$$\theta^* = \arg\min_{\theta} -\frac{1}{5}\sum_{k=1}^{5} \mathrm{F1}_k(\theta) \tag{11}$$

Where $F1_k(\theta)$ represents the F1-score for the k-th fold. Finally, the CatBoost model is constructed based on the Bayesian optimization results shown in Table(II), adopting the ordered boosting type, with LogLoss as the loss function, and the F1-score as both the evaluation metric and the early stopping criterion to avoid overfitting. After training, the model predicts the test data, generating anomaly probability

Model	β	γ	E & DG Learning_rate	D Learning_rate	latent_dim	batch_size	Epochs
VAE	0.1	None	0.0001	None	32	64	200
VAE-GAN	0.1	2	0.0001	0.0001	32	64	200

TABLE I

VAE & VAE-GAN GENERATOR HYPERPARAMETERS

scores and binary prediction results by applying a threshold of 0.5 to classify normal and anomalous instances. This trained model is stored for subsequent online real-time prediction.

E. Model Explainability (TreeSHAP)

Since CatBoost is based on tree-based model, it meets the conditions for deploying TreeSHAP, which effectively reduces computational complexity with detailed theory illustrated in Section(III-D). To further enhance the interpretability of anomalous samples, this study specifically selects 1,000 normal samples as background data, enabling Tree SHAP to highlight deviations of anomalous samples relative to normal samples. This dataset is preserved for subsequent online real-time interpretation tasks. Bar plot and Beeswarm plot are generated to illustrate feature importance rankings and contribution distributions across the dataset. Additionally, during online validation, a waterfall plot is employed to trace how each feature incrementally accumulates from the base value to the final predicted value, elucidating the reason for the anomaly.

V. EXPERIMENT METHODOLOGY

Supervised anomaly detection is often limited by class imbalance, leading to poor detection performance. This study employs VAE and VAE-GAN to generate anomalous samples, mitigate class imbalance, followed by CatBoost for anomaly detection and SHAP to interpret result. The evaluation framework is outlined below.

A. Data Analysis

Prior to model development, understanding data structure is the most fundamental and critical step in ML. EDA is employed to visually and statistically explore the processed data. Specifically, pie charts, bar plots, histograms, KDE plots, and KernelPCA are utilized to examine the proportions of minority anomalous samples and distribution differences between normal and anomalous samples.

B. Distance Metrics for Generated Anomalies Quality

To assess the similarity between anomalous samples generated by VAE and VAE-GAN and real anomalous samples, both models share the same neural network architecture and hyperparameters. Four statistical distance metrics —KL divergence, JS divergence, Wasserstein distance, and MMD —are adopted.

1) **KL Divergence :** Quantifies the asymmetric information difference between the real anomalous distribution and the generated anomalous distribution through the entropy difference derived from histograms, as show in Equation(12).

$$D_{KL}(P \,\|\, Q) = \sum P(x) \log \frac{P(x)}{Q(x)} \tag{12}$$

2) **JS Divergence :** A symmetric variant of KL divergence, providing a more balanced measure through the entropy difference of the average distribution, illustrated in Equation(13). Where $M = \frac{P+Q}{2}$ serves as the symmetric measure between the two distributions.

$$D_{JS}(P \,\|\, Q) = \frac{1}{2}\big[D_{KL}(P \,\|\, M) + D_{KL}(Q \,\|\, M)\big] \tag{13}$$

3) **Wasserstein Distance :** Capturing the geometric difference between distributions by optimizing the transportation costs from P to Q, expressed as Equation(14).

$$W(P, Q) = \min_{\gamma \in \prod} \sum_{x_p, x_q} \gamma(x_p, x_q) \|x_p - x_q\| \tag{14}$$

4) **MMD :** Calculates average difference between two distributions, utilizing the RBF kernel function to quantify the deviation between generated and real data in the feature space, as shown in Equation(15).

$$MMD(P, Q) = \sup_{\|f\|_{\mathcal{H}} \leq 1} \big(\mathbb{E}_p[f(x)] - \mathbb{E}_q[f(y)]\big) \tag{15}$$

Since KL and JS divergence primarily focus on the similarity of probability distribution and exhibit relatively low sensitivity to structural difference. Therefore, Wasserstein distance and MMD are introduced to account for deviations in spatial structure and feature space, achieving a more comprehensive evaluation of the realism of generated data in high-dimensional space. Finally, visualization using histograms and KDE is employed to present the distributions of real anomalous and generated anomalous samples, providing a more intuitive comparison.

C. Classification Metrics for Detection Performance

After utilizing VAE and VAE-GAN to generate anomalous samples, this study integrates the original data with the generated data and applies CatBoost to validate the detection performance. Evaluation metrics include Precision, Recall, F1-score, ROC-AUC, and PR-AUC. Additionally, a Confusion Matrix visually displays the model's classification results, including four categories: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). Here, TP indicates the model correctly classifies anomalies as anomalous, FP indicates misclassified normal samples as anomalous, FN indicates the model incorrectly classifies

Model	Max_depth	Iterations	Learning_rate	L2_leaf_reg	min_data_in_leaf
CatBoost	4	73	0.1	12	6

TABLE II
CatBoost Classifier Hyperparameters

anomalies as normal, and TN indicates the model correctly classifies normal samples as normal.

In anomaly detection, the model is expected to accurately identify anomalies (TP) while minimizing misclassification of normal samples as anomalies (FP), defined as Precision in Equation(16). Simultaneously, it should identify all anomalies while avoiding omission as normal (FN), defined as Recall in Equation(17). However, increasing recall may lead to numerous false positive, increasing human effort to screen misclassified samples. To balance precision and recall, the F1-score is introduced as their harmonic mean in Equation(18). ROC-AUC reflects the probability that the model assigns a higher score to an anomalous sample than a normal one, while PR-AUC focuses on the trade-off between precision and recall across thresholds.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{16}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{17}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

VI. Results

A. EDA

The UCI-SECOM dataset comprises 1,567 complete process records, each recording consists of 590 sensor measurements, labeled by the product's final test result (Pass/Fail). With real anomalous accounting for only 6% of the data, as shown in Fig.(2), the data exhibits significant class imbalance.

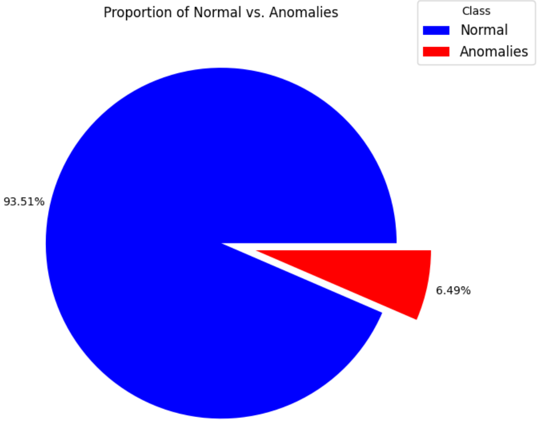

Fig. 2. Proportion of Normal vs. Anomalous Data

After completing the aforementioned data processing pipeline in Section(IV-B), the feature dimensionality was reduced from 590 to 33 (a 94.4% reduction). Subsequently, KernelPCA analysis, as shown in Fig.(3), revealed that the distributions of normal and anomalous data were not significantly distinct, violating the assumptions of unsupervised learning. Therefore, a supervised learning method was selected for this study. Next, real anomalous samples were extracted for minority anomaly synthesis based on VAE and VAE-GAN.

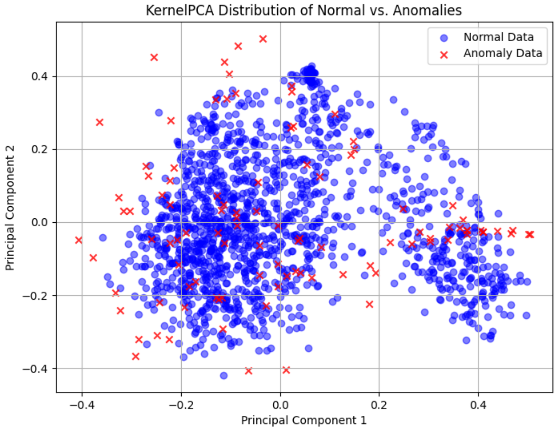

Fig. 3. KernelPCA Distribution of Normal vs. Anomalous Data

B. Anomalous Samples Generation

After extracting anomalous samples, these data were input into VAE and VAE-GAN models for anomaly data generation to address the class imbalance problem. First, the data generated by VAE is analyzed using histograms and KDE plots. Due to space limitations, this study only presents the generation results for the first four features, as shown in Fig.(4). The KDE plots of VAE-generated data appear more concentrated, exhibiting a narrower distribution. A further evaluation through a KernelPCA plot, showing in Fig.(5), also illustrates that the generated data does not adequately cover the real anomalous samples.

In contrast, the generation results of VAE-GAN are shown through histograms and KDE plots, similar to VAE, displaying only the first four features, as illustrated in Fig.(6). The KDE plots of the data generated by VAE-GAN more closely resemble the distribution of real anomaly patterns. Additionally, the KernelPCA plot in Fig.(7) shows that VAE-GAN achieves greater coverage of real anomalous data compared to VAE.

Subsequently, the distance metrics mentioned in Section(V-B) were utilized to quantify the distance between real

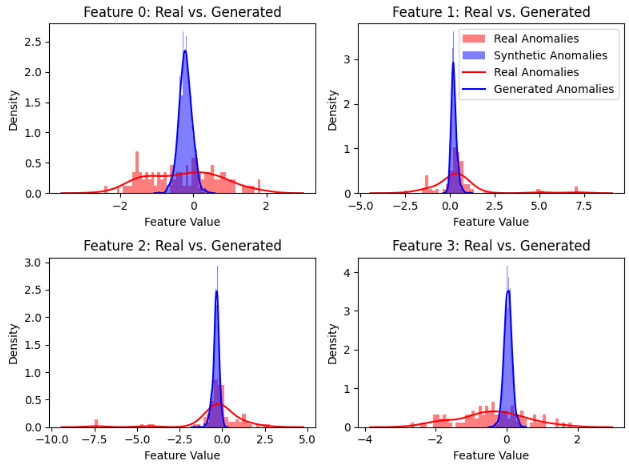

Fig. 4. KDE Distribution of Real vs. VAE-Generated Anomalies

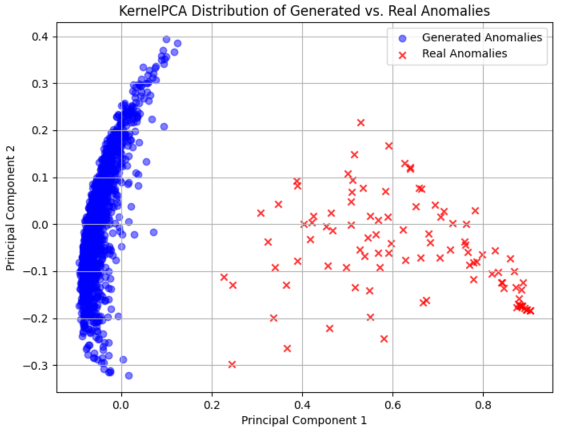

Fig. 5. KernelPCA Distribution of Real vs. VAE-Generated Anomalies

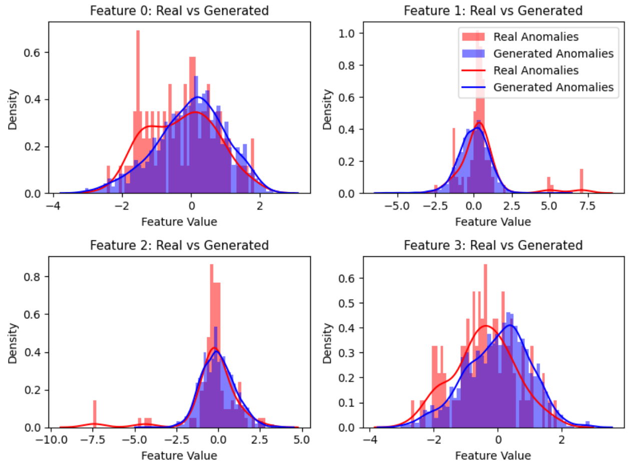

Fig. 6. KDE Distribution of Real vs. VAE-GAN-Generated Anomalies

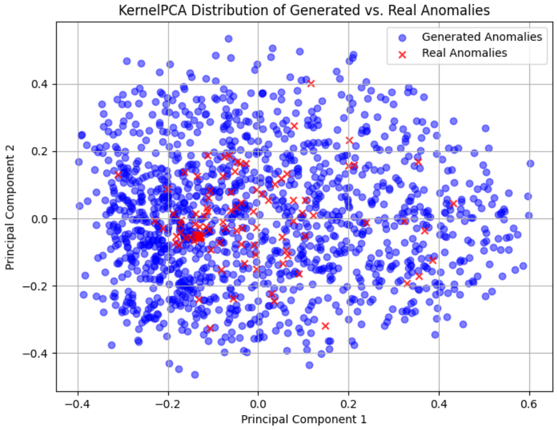

Fig. 7. KernelPCA Distribution of Real vs. VAE-GAN-Generated Anomalies

and generated anomalies, as shown in Fig.(8). Empirical results show that the distances between the VAE-GAN-generated and real anomalous data are smaller than those for VAE-generated data, conforming that VAE-GAN-generated data more closely aligns with the patterns of real anomalous data.

C. Anomaly Detection Performance

Based on the evaluation of the quality of generated anomalous data, it was confirmed that VAE-GAN outperforms VAE in terms of data quality. Next, this study further explores whether higher-quality generated data can enhance the performance of the CatBoost anomaly detection model. The anomalous data generated by VAE and VAE-GAN were combined with the original dataset to conduct anomaly classification tasks. Fig.(9) shows the classification results in a confusion matrix based on integrated VAE-generated data, while Fig.(10) displays the results for VAE-GAN-generated data. Consequently, VAE-GAN exhibits fewer false negatives and false positives, demonstrating that CatBoost, when combined with VAE-GAN, provides superior detection performance. Additionally, Table.(III) lists the anomaly detection performance results for the previous studies and the two data augmentation methods, VAE and VAE-GAN.

Furthermore, this study quantified the impact of different oversampling ratios on classification performance using VAE-GAN-generated anomalies. As shown in Fig.(11), without any data augmentation, the model failed to detect anomalies effectively due to the class imbalance issue. However, when the oversampling ratio was increased to 30%, the model's performance improved significantly, particularly in precision and recall. When the oversampling ratio reached 80%, the detection performance was comparable to that achieved at a 100% oversampling ratio. Therefore, the experiments have proven that anomaly detection performance can be effectively enhanced by increasing the oversampling ratio.

D. Feature Importance and Anomalous-Feature Analysis

To enhance the confidence of domain experts and assist them easier understanding anomaly points, this study introduced TreeSHAP as an XAI model. During the offline training phase, a dataset comprising normal samples was extracted as background data to highlight anomalous data

Model	Precision	Recall	F1-score	ROC-AUC	PR-AUC
SMOTE + DT (prior study)	0.869	0.958	0.911	None	None
SMOTE + AdaBoost (prior study)	0.919	0.926	0.965	None	None
SMOTE + GBT (prior study)	0.968	0.958	0.963	None	None
SMOTE + XGB (prior study)	0.975	0.958	0.966	None	None
VAE + CatBoost (proposed baseline)	0.981	0.949	0.965	0.986	0.989
VAE-GAN + CatBoost (proposed improvement)	0.988	0.953	0.97	0.985	0.989

TABLE III
COMPARISON OF DETECTION PERFORMANCE

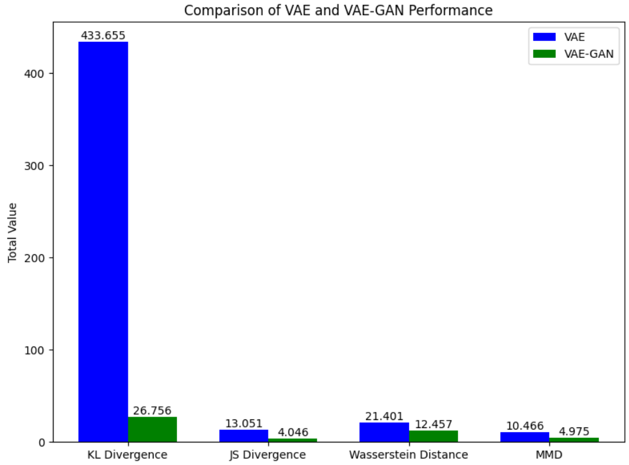

Fig. 8. Distance Between Real vs. VAE/VAE-GAN Generated Anomalies

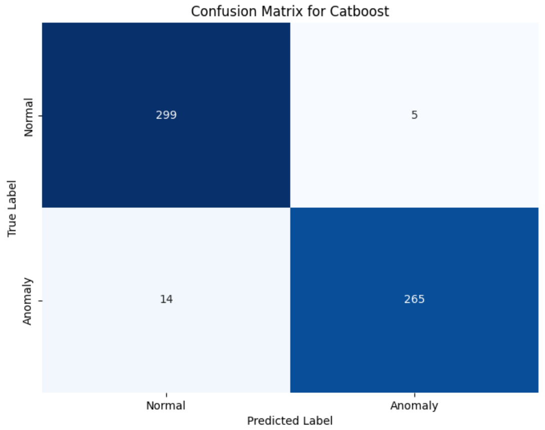

Fig. 9. CatBoost Performance with VAE-generated Anomalies

and serve as a reference foundation for subsequent online real-time validation. Initially, this study employed bar plot to present the average SHAP value of each feature, enabling the analysis of feature importance. As shown in Fig.(12), the most significant feature was identified as feature 511, indicating its role as the primary factor contributing to anomalies. However, in real industrial settings, a sequential relationship among features may exist, where anomalies in earlier feature values could trigger subsequent anomalies. Therefore, to further assist domain experts in identifying the earliest anomaly point, this study analyzed the relationship

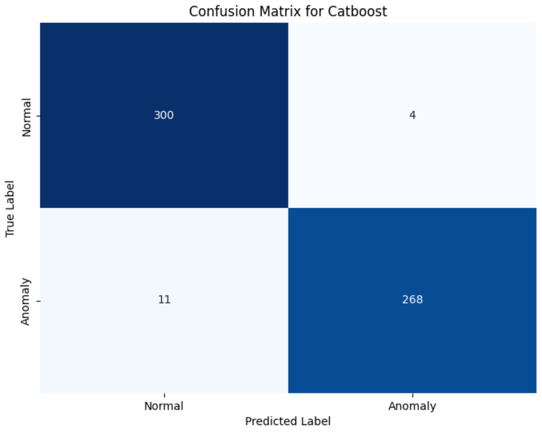

Fig. 10. CatBoost Performance with VAE-GAN-generated Anomalies

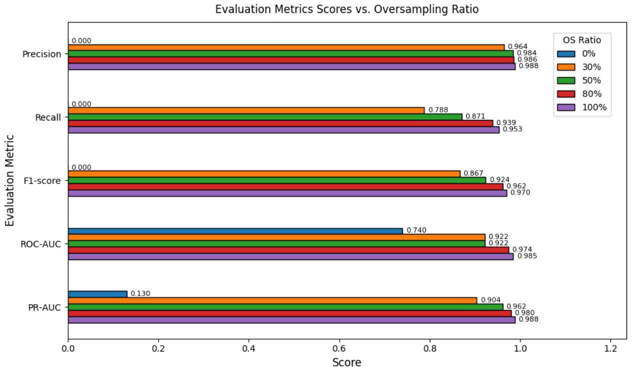

Fig. 11. CatBoost detection result vs. oversampling ratio

between ordered feature values and anomaly induction, visualized through a beeswarm plot. As demonstrated in Fig.(13), for instance, lower value of Feature 14 or higher value of Feature 21 yield higher SHAP values, gradually leading to subsequent anomalies.

Finally, online simulated streaming data was employed to evaluate the practical performance of the anomaly detection model. This study adopted waterfall plots to provide detailed explanations of anomalous features within the data. As illustrated in Fig.(14), an anomaly instance was influenced by Feature 59, with a value of 1.962, significantly driving its classification as anomalous. This finding aligns with the beeswarm plot presented in Fig.(13), indicating that excessively high value of Feature 59 tends to produce higher

SHAP value, ultimately leading to an anomaly.

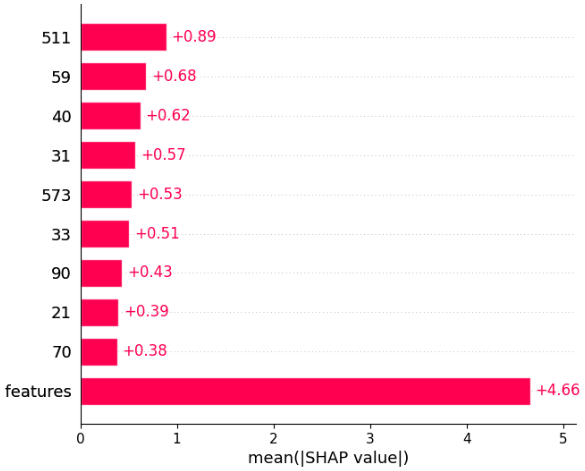

Fig. 12. Feature Importance Analysis

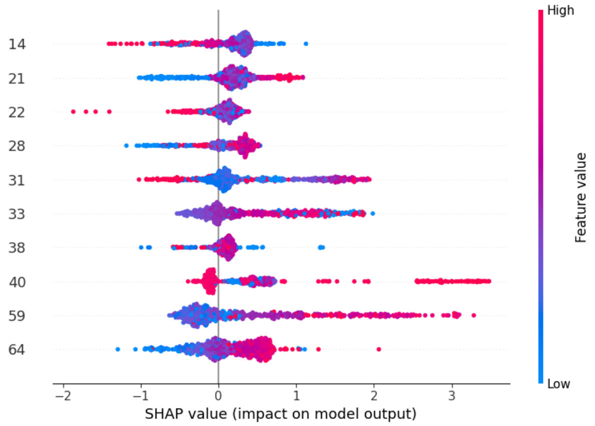

Fig. 13. Relationships between Feature Value and SHAP Value

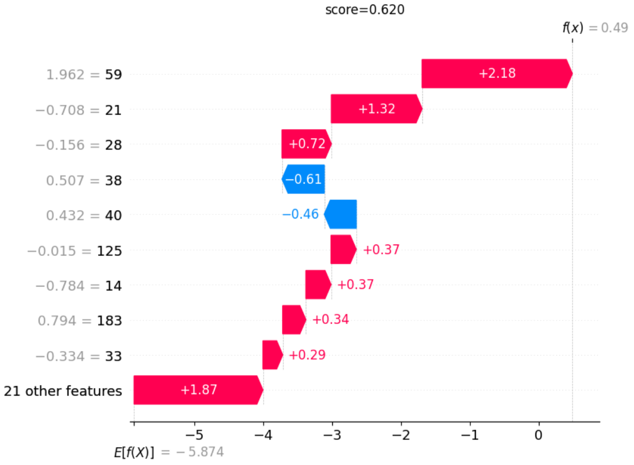

Fig. 14. Anomalous Feature Analysis for a Detected Anomaly Instance

VII. Discussion and Conclusion

A. Discussion & Conclusion

This study's VAE-GAN data augmentation effectively tackles class imbalance in anomaly detection for industrial settings, enabling CatBoost to achieve 98% precision and 96% recall. TreeSHAP enhances interpretability, and the pipeline is validated via online simulated data streams. It outperforms prior SMOTE-based tree models on the UCI-SECOM dataset, as shown in Table(III). This is the first study to integrate data augmentation, anomaly detection, and interpretability into a complete pipeline, aiding domain experts in quickly identifying production line anomalies.

For Aim(1), this study applies EDA-driven preprocessing to handle missing values. Low-variance features and those weakly correlated with the label are removed, and only one representative is kept from highly correlated pairs. These steps lower computational cost and preserve key features. However, the process relies on linear assumptions and fixed thresholds, limiting generalization in imbalance, nonlinear settings where rare but critical features are discarded. To address this, adopting nonlinear dimensionality reduction methods (e.g., t-SNE, UMAP) may help capture complex patterns.

For Aim(2), VAE-GAN combines VAE latent space regularization with GAN's adversarial training, generating more diverse and realistic anomalies than VAE and SMOTE. However, this study revealed VAE-GAN tends to produce data distributions centered around real anomaly points, potentially indicating mode collapse, where the model captures only certain peripheral modes while missing primary ones, as shown in Fig.(15). To address this, VAE-GAN hyperparameters should be tuned, such as lowering β from 1.0 to 0.1 and increasing γ from 1 to 2 to improve coverage.

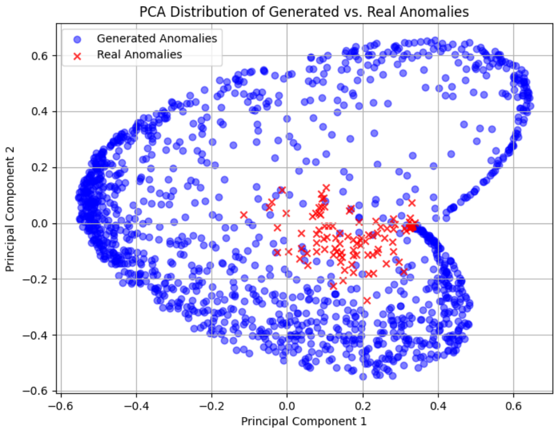

Fig. 15. Tendency of VAE-GAN mode collapse

For Aim(3), although VAE-GAN generates higher-quality anomalies than VAE, CatBoost results show little improvement from VAE-GAN augmentation. Two factor may explain this: (i) the number of generated anomalies was insufficient to shift CatBoost's decision boundary, and (ii) CatBoost did not fully exploit the generated samples. Thus, for small dataset, VAE may suffice with CatBoost for anomaly detection.

For Aim(4), increasing the oversampling ratio improved detection performance, with 100% achieving the best overall metrics. However, fixing the ratio at 100% may introduce unnecessary computational cost, hindering real-world deployment. Thus, an adaptive scheme for selecting the ratio could improve both efficiency and generalization.

For Aim(5), TreeSHAP highlights feature importance, links between feature values and anomalies, and instance-level attributions, improving interpretability. However, as this dataset is public, validation by domain experts was not possible. Future work should deploy the model in industrial settings and collaborate with domain experts to confirm interpretability.

For Aim(6), online evaluation revealed one missed detection, likely due to CatBoost's overfitting on small datasets, which resulted in poor generalization, though this study had already applied basic strategies to alleviate it. A possible solution is more meticulous hyperparameters tuning of CatBoost to further mitigate overfitting.

Future research directions could involve exploring diffusion models as generative models to overcome the mode collapse issue potentially occurring in GAN-based model. Additionally, it is recommended to deploy this model on different production lines for anomaly detection to validate its generalization capability. These efforts can overcome current limitations, thereby enhancing the anomaly detection system's robustness and versatility.

REFERENCES

[1] C.-F. Chien, T. Hong Van Nguyen, Y.-C. Li, and Y.-J. Chen, "Bayesian decision analysis for optimizing in-line metrology and defect inspection strategy for sustainable semiconductor manufacturing and an empirical study," *Computers Industrial Engineering*, vol. 182, p. 109421, 2023.

[2] P. Kamat and R. Sugandhi, "Anomaly detection for predictive maintenance in industry 4.0-a survey," *E3S Web of Conferences*, vol. 170, 05 2020.

[3] H. Rostami, J. Blue, and C. Yugma, "Automatic equipment fault fingerprint extraction for the fault diagnostic on the batch process data," *Applied Soft Computing*, vol. 68, pp. 972–989, 2018.

[4] B. K. Iwana and S. Uchida, "An empirical survey of data augmentation for time series classification with neural networks," *PLoS ONE*, vol. 16, 2020.

[5] Z. Dai, L. Zhao, K. Wang, and Y. Zhou, "Mode standardization: A practical countermeasure against mode collapse of gan-based signal synthesis," *Applied Soft Computing*, vol. 150, p. 111089, 2024.

[6] B. An, S. Wang, F. Qin, Z. Zhao, R. Yan, and X. Chen, "Adversarial algorithm unrolling network for interpretable mechanical anomaly detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, pp. 6007–6020, 2023.

[7] S. Kricheff, E. Maxwell, C. Plaks, and M. Simon, "An explainable machine learning approach for anomaly detection in satellite telemetry data," *2024 IEEE Aerospace Conference*, pp. 1–14, 2024.

[8] P. Mathur. (2008) Uci secom dataset. [Online]. Available: https://www.kaggle.com/datasets/paresh2047/uci-semcom/data

[9] A. A. Nuhu, Q. Zeeshan, B. Safaei, and M. A. Shahzad, "Machine learning-based techniques for fault diagnosis in the semiconductor manufacturing process: a comparative study," *The Journal of Supercomputing*, vol. 79, pp. 2031–2081, 2022.

[10] Y. Wang, G. Huang, S. Song, X. Pan, Y. Xia, and C. Wu, "Regularizing deep networks with semantic data augmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 3733–3748, 2020.

[11] X. Ma and W. Shi, "Aesmote: Adversarial reinforcement learning with smote for anomaly detection," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 943–956, 2021.

[12] S.-K. S. Fan, D.-M. Tsai, and P.-C. Yeh, "Effective variational-autoencoder-based generative models for highly imbalanced fault detection data in semiconductor manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 36, no. 2, pp. 205–214, 2023.

[13] J. Chung, B. Shen, and Z. Kong, "Anomaly detection in additive manufacturing processes using supervised classification with imbalanced sensor data based on generative adversarial network," *Journal of Intelligent Manufacturing*, pp. 1–20, 2022.

[14] J. Shabbir and W. H. Awan, "An efficient shewhart-type control chart to monitor moderate size shifts in the process mean in phase ii," *Quality and Reliability Engineering International*, vol. 32, pp. 1597 – 1619, 2016.

[15] D.-H. Hoang and H. Nguyen, "A pca-based method for iot network traffic anomaly detection," *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pp. 381–386, 2018.

[16] D. Romeres, D. K. Jha, W. Yerazunis, D. Nikovski, and H. A. Dau, "Anomaly detection for insertion tasks in robotic assembly using gaussian process models," *2019 18th European Control Conference (ECC)*, pp. 1017–1022, 2019.

[17] A. Robles-Durazno, N. Moradpoor, J. McWhinnie, and G. Russell, "A supervised energy monitoring-based machine learning approach for anomaly detection in a clean water supply system," *2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pp. 1–8, 2018.

[18] A. Gerling, H. Ziekow, A. Hess, U. Schreier, C. Seiffer, and D. Abdeslam, "Comparison of algorithms for error prediction in manufacturing with automl and a cost-based metric," *Journal of Intelligent Manufacturing*, vol. 33, pp. 555 – 573, 2022.

[19] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, pp. 15:1–15:58, 2009.

[20] M. Jeon, I.-H. Choi, S.-W. Seo, and S.-W. Kim, "Extremely rare anomaly detection pipeline in semiconductor bonding process with digital twin-driven data augmentation method," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 14, no. 10, pp. 1891–1902, 2024.

[21] T. Chen, X. Liu, B. Xia, W. Wang, and Y. Lai, "Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder," *IEEE Access*, vol. 8, pp. 47 072–47 081, 2020.

[22] H. Kayan, R. Heartfield, O. Rana, P. Burnap, and C. Perera, "Real-time anomaly detection for industrial robotic arms using edge computing," *IEEE Internet of Things Journal*, vol. 12, no. 15, pp. 29 696–29 712, 2025.

[23] S. Nazat, O. Arreche, and M. Abdallah, "On evaluating black-box explainable ai methods for enhancing anomaly detection in autonomous driving systems," *Sensors (Basel, Switzerland)*, vol. 24, 2024.

[24] L. Antwarg, R. M. Miller, B. Shapira, and L. Rokach, "Explaining anomalies detected by autoencoders using shapley additive explanations," *Expert Systems with Applications*, vol. 186, p. 115736, 2021.

[25] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv*, 2022.

[26] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International conference on machine learning*. PMLR, 2016, pp. 1558–1566.

[27] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[28] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," *Advances in neural information processing systems*, vol. 31, 2018.

[29] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[30] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," *arXiv preprint arXiv:1802.03888*, 2018.