From Raw to Ready: Industrial Fault Data Enhancement Via Preprocessing and Balancing

Suroor M. Albattat
College of Engineering, Al-Iraqi University, Saba'a Abkar Complex, Baghdad, Iraq
sroor.m.taher@aliraqia.edu.iq (corresponding author)

Baraa M. Albaker
College of Engineering, Al-Iraqi University, Saba'a Abkar Complex, Baghdad, Iraq
baraamalbaker@gmail.com

Malik A. Alsaedi
College of Engineering, Al-Iraqi University, Saba'a Abkar Complex, Baghdad, Iraq
maliksaady@yahoo.com

Received: 16 June 2025 | Revised: 22 July 2025 | Accepted: 11 August 2025

 DOI: https://doi.org/10.48084/etasr.12784

ABSTRACT

In recent years, predictive maintenance has emerged as a critical component for improving the efficiency and reliability of industrial systems. However, much of the existing research has primarily emphasized model development, often overlooking the fundamental role of data quality and class distribution in shaping predictive performance. To address this gap, this study proposes an integrated preprocessing framework that ensures high-quality data readiness across all stages. A case study was conducted on a dataset of industrial sensors for fault prediction. The preprocessing pipeline involved handling missing values using K-Nearest Neighbors (KNN), detecting outliers with Isolation Forest (IF), and correcting abnormal values through the Clipping method. To address data imbalance, synthetic data were generated using Generative Adversarial Networks (GAN), Variational Autoencoders (VAE), and a hybrid GAN-VAE model that leverages the strengths of both approaches. The hybrid GAN-VAE demonstrated superior data generation performance, yielding the highest Pearson correlation and best Kernel Density Estimation (KDE) fit, thereby ensuring dataset reliability for training. The effectiveness of the preprocessing framework was validated using a 1-Dimensional Convolutional Neural Network (1D-CNN) classifier, which achieved a high accuracy of 98.83%.

Keywords-data preprocessing; imbalanced data; machine learning; outliers; Generative Adversarial Network (GAN)

I. INTRODUCTION

Technological advancements and the development of Machine Learning (ML) models have significantly improved fault diagnosis in industrial devices and equipment, enabling more accurate predictions [1]. However, ML techniques are highly sensitive to the distribution of training data, particularly in classification tasks [2]. Imbalanced datasets often bias predictions toward the majority class, reducing model reliability and leading to misclassification of minority class samples [3-5]. Two main strategies are commonly used to address imbalance: undersampling, which reduces majority samples but risks information loss, and oversampling, which generates additional samples for the minority class to achieve balanced distribution [6].

It is also common practice for preprocessing to precede imbalance handling, since missing values can substantially affect ML model accuracy [7]. For instance, the K-Nearest Neighbors (KNN) method is effective for addressing missing data, as it leverages neighborhood similarity to estimate values with higher accuracy compared to simple deletion or replacement with mean/median [8]. In addition, handling extreme values is also crucial for the preprocessing phase, as they often arise due to mechanical disturbances or operational changes in industrial settings [9, 10]. Detecting these values using Isolation Forest (IF) is particularly efficient, as it isolates anomalies through decision tree ensembles [11]. To manage these abnormal values, the clipping method offers a straightforward solution by setting upper and lower bounds without discarding data [12].

This study utilizes data from the IMPROVE project, collected from a non-woven factory [13], comprising measurements from vibration, temperature, pressure, and current sensors. The dataset is highly imbalanced, with 228,416 healthy samples and only 8 faulty samples, necessitating

effective resampling strategies. Among oversampling approaches considered for this task were Generative Adversarial Networks (GANs), a widely used approach that consists of a generator that synthesizes data and a discriminator that evaluates realism against true data [14, 15]. Another approach considered was the Variational Autoencoder (VAE) [16, 17], although comparative studies indicate that GANs generally outperform VAEs in data augmentation [18, 19].

To overcome the limitations of using a standalone GAN or VAE, this study proposes a hybrid GAN-VAE model that integrates the advantages of both methods. Synthetic data generated by GAN, VAE, and GAN-VAE were compared using Pearson's correlation coefficient and Kernel Density Estimation (KDE) plots, with the hybrid approach showing superior alignment with real data. This demonstrates enhanced data quality and greater potential for model generalization in industrial fault detection. To validate preprocessing effectiveness, a 1-Dimensional Convolutional Neural Network (1D-CNN) classifier was trained, achieving very high accuracy and confirming the robustness of the proposed framework.

II. RELATED WORKS

Data preprocessing is a pivotal step in preparing raw data for analysis and prediction, ensuring reliability in system design and data-driven control. Seceral studies have emphasized its importance, focusing on techniques for handling missing values and detecting outliers, both of which strongly affect model efficiency [20]. For example, the KNN method has been widely applied due to its high effectiveness in estimating missing data [21]. Authors in [22] highlighted the necessity of preprocessing to improve sensor modeling for real-time fault detection. Similarly, KNN has been applied to detect and process missing values, while methods such as Interquartile Range (IQR) and IF have been used for outlier detection [23].

Additionally, data imbalance in large datasets also poses challenges in classification, particularly in artificial error detection. Oversampling methods such as GAN and VAE have been employed to generate artificial samples for the minority class, ensuring a more balanced class distribution and more reliable model performance [13, 17, 18, 24, 25]. However, most existing works focus on isolated preprocessing techniques rather than an integrated pipeline.

III. DATASET DESCRIPTION

This study utilizes a publicly available dataset introduced in [26], accessible via Kaggle. The dataset originates from the IMPROVE project and was collected from a non-wovens factory. It contains 228,614 intact samples and only 8 defective samples, presenting a highly imbalanced distribution. The dataset comprises multivariate time-series sensor measurements from industrial machines that monitor physical and operational parameters, including:

- Vibration sensors (L_i) capturing mechanical anomalies.
- Temperature sensors (A_i) monitoring component temperatures.
- Pressure sensors (B_i) monitoring pressure conditions.
- Current/Voltage sensors (C_i) monitoring electrical characteristics.

Together, these signals provide a continuous record of machinery behavior, enabling predictive maintenance, anomaly detection, and fault prediction.

The dataset was partitioned by class to ensure that all classes were reflected in a balanced manner. It was divided into three subsets: 70% for training, 15% for validation, and 15% for testing using a random constant value (random_state=42).

IV. MISSING VALUE HANDLING

Missing values were addressed using the KNN imputer, which has been shown to outperform traditional methods such as mean and median replacement in terms of accuracy across various domains [21]. We applied KNN with K = 5, achieving low Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) values, making it a robust choice for maintaining dataset integrity. Other parameters used are presented in Table I. Figure 1 illustrates the distribution of missing values across features before imputation, with each feature containing 12 missing entries. The KNN pseudocode algorithm used is presented below:

```
Algorithm 1: Missing values imputation
using KNN
Input: Dataset 𝒳 with missing values
Output: New dataset 𝒳_new
   1. For each row 𝑖 in 𝒳
   2. For each feature 𝑗 in 𝑖 that has a
      missing value:
   3. Identify the KNN of the instance 𝑖
         Compute Euclidean distance using
         available (non-missing) features
   4. Estimate the missing value 𝑥_ij:
         𝑥_ij = (1/k) · sum of 𝑥_j value from
         the K-neighbors
   5. Store the updated instance in a new
      dataset
Return 𝒳_new
```

TABLE I. KNN IMPUTER PARAMETERS

#	Parameter	Value	Description
1	K-neighbors	5	Number of nearest neighbors used for imputation
2	Weights	uniform	Equal weight assigned to all neighbors (default)
3	Metric	nan_euclidean	Distance metric ignoring missing values during computation

V. OUTLIER DETECTION

Outliers were detected using the IF algorithm, which is highly efficient for high-dimensional industrial datasets. This approach is particularly suitable for sensor data, as demonstrated in [27], where an enhanced IF variant was proposed for improved anomaly detection in industrial environments. The method isolates anomalous readings that could otherwise distort learning processes. Table II presents the

parameters used. Figure 2 presents the distribution of outliers detected in this dataset. The IF pseudocode algorithm used is presented below:

```
Algorithm 2: Outlier detection using IF
Initialization:
    Load dataset 𝒳
    Set parameters: n_estimators(T),
    contamination rate (θ), random_state
For each sample x in 𝒳:
    Build T isolation trees
    Compute average path length h(x)
    Compute anomaly score:
        s(x) = 2^(h(x)/ c(n))
            if s(x) > θ:
            label x = 1 (outlier)
                label x = 0 (normal)
Return labeled dataset
```

Where: $\mathcal{X} \rightarrow input\ dataset;\ x \rightarrow single\ data\ point;$ $T \rightarrow number\ of\ trees;\ h(x) \rightarrow avg\ path\ legth\ of\ x;$ $s(x) \rightarrow outlier\ score\ of\ x;\ \theta \rightarrow threshold\ of\ outliers$

TABLE II. IF PARAMETERS

#	Parameter	Value	Description
1	N-estimators	100	Number of base estimators (trees)
2	Contamination	0.01	Expected outlier ratio in the dataset
3	Max_samples	auto	Number of samples drawn per tree (default)
4	Random_state	42	Reproducibility controln
5	Behaviour	default	Modern scikit-learn setting (no modification needed)

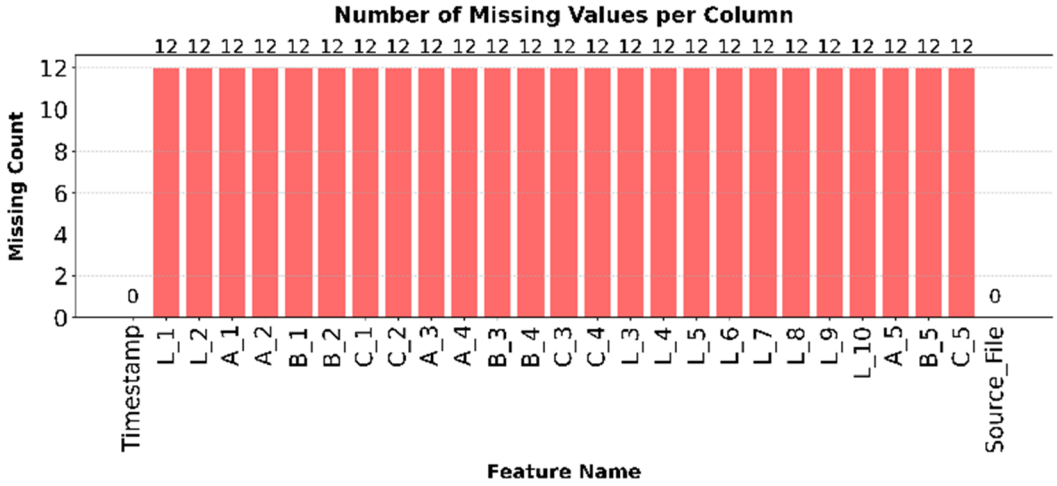

Fig. 1. Visualization of missing value before applying KNN imputation.

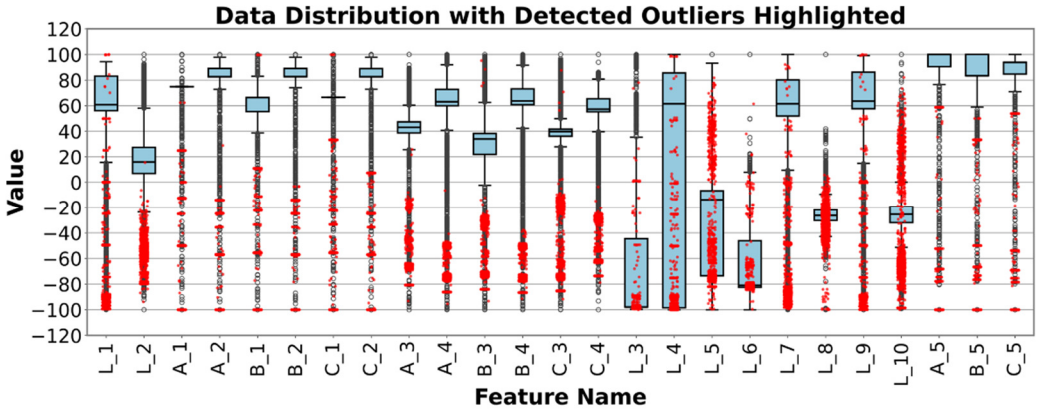

Fig. 2. Data distribution after outlier detection. (Red dots represent outliers detected by the IF method in the dataset).

VI. OUTLIER HANDLING

After detecting outliers, the extreme values were processed by replacing them with the median of the corresponding feature. This approach is widely adopted because the median represents the central tendency of the data and is not affected by extreme values [28]. Thus, it ensures that the data distribution is preserved while minimizing the influence of anomalies. The parameters used are listed in Table III. Figure 3 illustrates the dataset after outlier processing. The pseudocode algorithm used is presented below:

```
Algorithm 3:
Initialize:
```

Load dataset X
For each numeric column j:
    Compute median m_j =median (X_j)
For each sample X_{ij}:
    If X_{ij} is an outlier:
        Replace $X_{ij} = m_j$
    Else:
        Keep X_{ij} unchanged
Return the new dataset
Where: $X_{ij} \rightarrow$ *feature j in sample I*; $m_j \rightarrow$ *median of column j*; $X_j \rightarrow$ *all values in column j*

TABLE III. OUTLIER HANDLING PARAMETERS

#	Parameter	Value	Description
1	Outlier column	0 (normal), 1 (abnormal)	Used to identify outliers before processing
2	Replacement value	Mean	Replace detected outliers with the column median

VII. ABNORMAL VALUE HANDLING

A. IQR-Based Abnormal Detection and Median Imputation

To further enhance data robustness and minimize the influence of abnormal values, the IQR method was applied. This approach defines lower and upper thresholds based on the data's quartiles and identifies values outside this range as abnormal. These values were then replaced with the median of the respective feature. This ensures that the overall structure of the data is preserved while reducing the skew introduced by extreme deviations. Previous studies [29] have confirmed that IQR-based filtering enhances model robustness. The lower and upper thresholds are defined as:

$$IQR = Q_3 - Q_1 \quad (1)$$

$$Lower\ bound = Q_1 - 1.5 \cdot IQR \quad (2)$$

$$Upper\ bound = Q_3 + 1.5 \cdot IQR \quad (3)$$

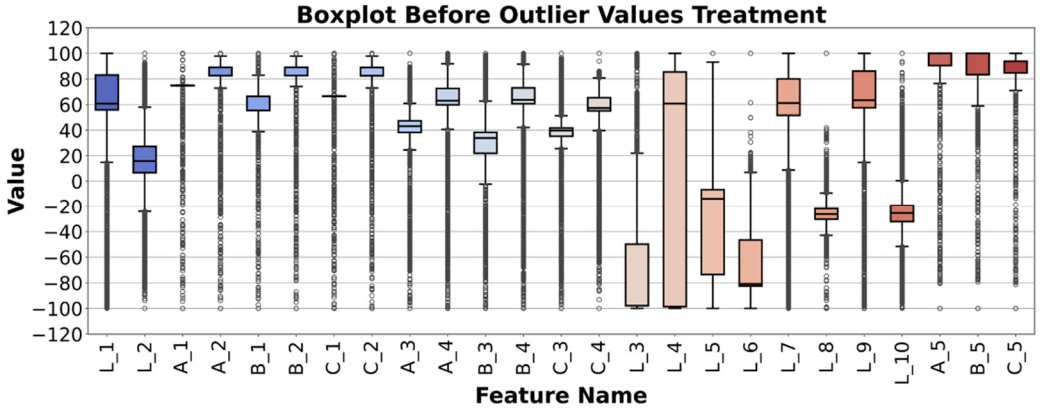

Fig. 3. Boxplot after processing extreme values using the median (where the outliers were replaced by the median for each feature, which reduced the data dispersion).

Values lying outside this range are treated as abnormal and replaced with the column median. Table IV lists the parameters used for the IQR method. Figure 4 shows the distribution of data after processing abnormal values, highlighting the reduced spread of extreme points, particularly on the right tail. The IQR method pseudocode algorithm used is presented below:

Algorithm 4: Abnormal values handling using IQR and median imputation
Initialization:
Read dataset X
For each numeric column X_j in X:
    Calculate $Q_1 = 25th\ percentile\ of\ X_j$
    Calculate $Q_3 = 75th\ percentile\ of\ X_j$
    Calculate $IQR = Q_3 - Q_1$
    Define Lower bound $= Q_1 - 1.5 \cdot IQR$
    Define Upper bound $= Q_3 + 1.5 \cdot IQR$
For each $x\ in\ X_j$:
    If $x < lower\ or\ x > upper$:
        Replace $x = median(X_j)$
Return a new dataset

TABLE IV. IQR-BASED OUTLIER DETECTION AND MEDIAN IMPUTATION PARAMETERS

#	Parameter	Description
1	Q_1	25th percentile of the column
2	Q_3	75th percentile of the column
3	IQR	$Q_3 - Q_1$
4	Threshold	$\pm 1.5 \cdot IQR$
5	Replacement	Median of the column

B. IQR-Based Clipping Method

An alternative approach adopted in this study for handling abnormal values was the IQR-based clipping method. Abnormal values, which may distort model performance, were detected using the IQR approach. Specifically, values lower than $Q_1 - 1.5 \cdot IQR$ or higher than $Q_3 \cdot 1.5\, IQR$ were considered abnormal. Instead of deleting these values or replacing them with central tendency measures, clipping was applied by truncating extreme values to the nearest valid limit

(lower or upper bound). This approach preserves the dataset size while reducing the impact of extreme deviations.

The method has been successfully adopted in various fields, including image denoising, where it has been shown to outperform traditional median filtering [30]. The parameters used for this method are presented in Table V. The results presented in Figure 5 demonstrate reduced data distortion and improved robustness compared to median replacement.

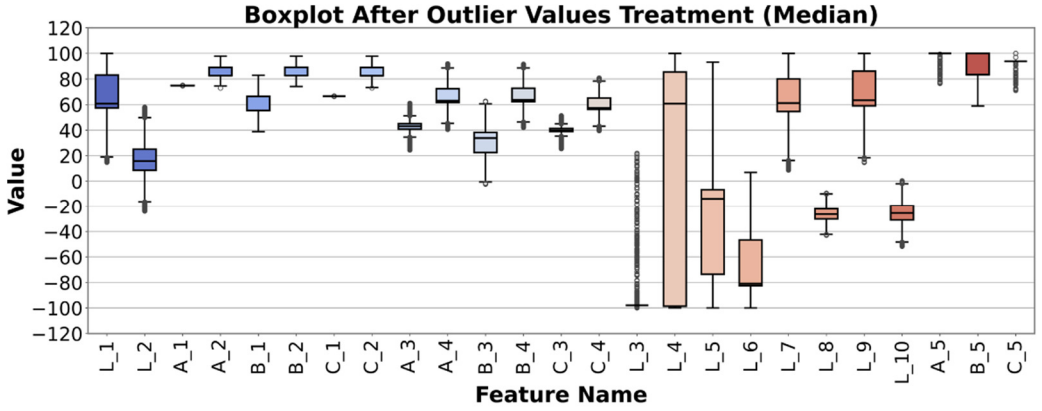

Fig. 4. Boxplot after abnormal value treatment (median).

The pseudocode algorithm describing this method is presented below:

```
Algorithm 5: Abnormal values handling
using IQR and clipping imputation
Initialization:
Load dataset 𝒳
For each feature 𝒳_j in 𝒳:
        Q_1 = 25th percentile (𝒳_j)
        Q_3 = 75th percentile (𝒳_j)
        IQR = Q_3 − Q_1
        Set Lower = Q_1 − 1.5 · IQR
        Set Upper = Q_3 + 1.5 · IQR
For each value x_ij in column 𝒳_j
        If  x_ij < Lower:
            x_ij = Lower
        Else if x_ij > Upper
            x_ij = Upper
        Else:
            keep x_ij as it is
Return the new dataset
```

TABLE V. IQR-BASED CLIPPING PARAMETERS

#	Parameters	Description
1	Q_1	25th percentile of the column
2	Q_3	75th percentile of the column
3	IQR	$Q_3 - Q_1$
4	Lower bound	$Q_1 - 1.5 \cdot IQR$
5	Upper bound	$Q_3 + 1.5 \cdot IQR$
6	Replacement	Extreme values replaced by nearest bound (clip)

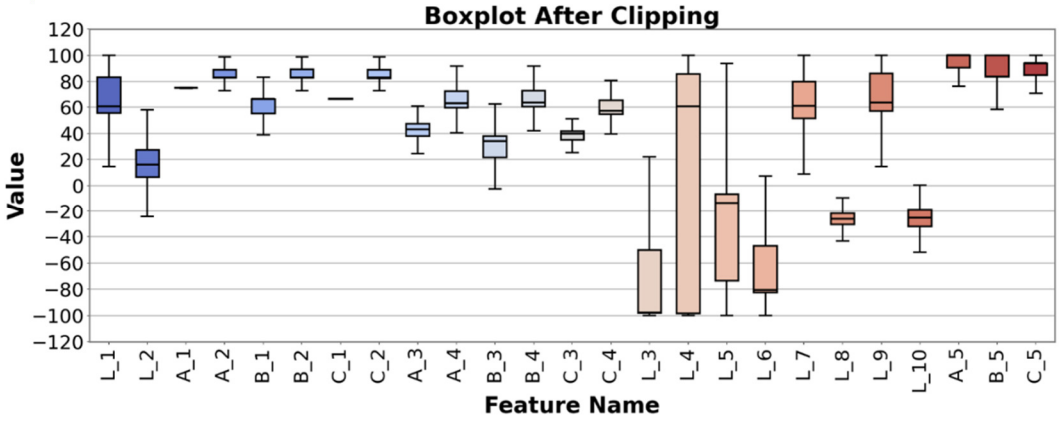

Fig. 5. The boxplot shows data distribution after clipping the abnormal values and improving data alignment with IQR.

VIII. DATA BALANCING

A. Generative Adversarial Network (GAN)

GANs provide a flexible approach for generating synthetic data, particularly in scenarios where minority classes are severely underrepresented. GAN consists of two competing networks:

- The generator, which produces synthetic samples from random noise.
- The discriminator, which distinguishes between real and generated samples.

During training, the generator improves its ability to produce realistic data, while the discriminator enhances its ability to detect synthetic data. Once trained, the generator produces synthetic faulty samples in proportion to the number of healthy samples, thereby achieving data balance [31]. The GAN parameters used are listed in Table VI. The pseudocode algorithm describing GAN is presented below:

```
Algorithm 6: Data balancing using GAN
Initialization:
Load dataset 𝒳
Normalize data: 𝒳_norm = (𝒳 − μ)/ σ
Training:
   For each epoch, do:
     Sample random noise vector: 𝒵 ~ 𝒩(0,1)
     Generate fake data:  𝒳_fake = G(𝒵)
     Generate discriminator output:
           D_real = D(𝒳_real)
           D_fake = D(𝒳_fake)
     Compute losses:
           L_D = −log(D_real) − log (1 − D_fake)
           L_G = −log(D_fake)
Update discriminator and generator
After training:
       Generate new data: 𝒳_gen = G(𝒵)
       Inverse transform: 𝒳_final = 𝒳_gen · σ + μ
Return a balanced dataset
```

TABLE VI. GAN PARAMETERS

#	Parameter	Value
1	Latent dimension	64
2	Generator layers	[64, 128, 256, output]
3	Discriminator layers	[256, 128, 1]
4	Activation function	LeakyReLU
5	Epochs	3000
6	Batch size	32
7	Optimizer	Adam
8	Learning rate	0.0001

B. Variational Autoencoders (VAE)

VAEs provide another generative solution for handling class imbalance. Unlike GANs, VAEs explicitly learn a latent probability distribution of the input data. They consist of:

- An encoder, which compresses the input into a latent distribution defined by μ and σ^2.
- A decoder, which reconstructs the input from latent samples.

Once trained, new synthetic samples are generated by sampling from the latent distribution and decoding. The training objective combines Mean Squared Error (MSE) with a Kullback-Leibler (KL) divergence for regularization, ensuring both accuracy and diversity in generated samples [32]. The VAE parameters used are presented in Table VII. The VAE pseudocode algorithm used is presented below:

```
Algorithm 7: Data balancing using VAE
Initialization:
Load dataset 𝒳
Normalize data: 𝒳_norm = (𝒳 − μ)/ σ
Training Loop (for each epoch):
   Encode input:
      μ, log σ² = Encoder(𝒳_norm)
      Sample noise: ε ~ 𝒩(0,1)
      Compute latent vector:
      𝒵 = 𝒵_mean + ε · exp(0.5 · 𝒵_logvar)
   Decode:
       Reconstruct data: 𝒳̂ = Decoder(𝒵)
   Compute Loss:
       MSE = ∑(𝒳_i − 𝒳̂_I)²
       KL = −0.5 · ∑(1 + log(σ²)) − μ² − σ²
       L = MSE + KL
After training:
       Sample new latent vector: 𝒵 ~ 𝒩(0,1)
       Generate synthetic data: 𝒳_gen =
       Decoder(𝒵)
       Inverse normalization: 𝒳 = 𝒳_gen · σ + μ
Return new balanced data
```

TABLE VII. VAE PARAMETERS

#	Parameter	Value
1	Latent dimension	16
2	Hidden layers	[64] One layer of 64 neurons
3	Activation function	LeakyReLU is used in the encoder and decoder
4	Epochs	3000
5	Batch size	32
6	Optimizer	Adam (lr=0.0001)
7	Loss function	MSE+ KL Divergence

C. Proposed Model GAN-VAE

The proposed hybrid GAN-VAE model integrates the reconstruction ability of VAE with the adversarial training mechanism of GAN. By combining these strengths, the GAN-VAE produces synthetic data that is both structurally consistent and highly realistic. The parameters used are listed in Table VIII.

TABLE VIII. GAN-VAE MODEL PARAMETERS

#	Parameter	Value
1	Latent dimension	16
2	Hidden layers	[64], [64]
3	Activation function	LeakReLU (Encoder), LeakyReLU(Discriminator)
4	Epochs	2500
5	Batch size	32
6	Optimizer(G/E)	Adam (0.0001, 0.0005)
7	Optimizer(D)	Adam (0.001)
8	Loss function	MSE+ KL

The GAN-VAE pseudocode algorithm used is presented below:

Algorithm 8: Data balancing using GAN-VAE

Initialize:
Load dataset $\mathcal{X}$
Normalize data: $\mathcal{X}_{norm} = (\mathcal{X} - \mu)/\sigma$
Encoder:
$\sigma, \mu = Encoder\ (\mathcal{X}_{norm})$
Sample noise: $\varepsilon \sim \mathcal{N}(0,1)$
Latent vector: $\mathcal{Z} = \mu + \varepsilon \cdot \sigma$
Decoder/ Generator:
Reconstructed data: $\hat{\mathcal{X}} = Generator\ (\mathcal{Z})$
Discriminator:
$D_{real} = Discriminator\ (\mathcal{X}_{norm})$
$D_{fake} = Discriminator(\hat{\mathcal{X}})$
Losses:
GAN loss: $L_{GAN} = -\log D_{fake}$
Reconstruction loss: $L_{recon} = || \mathcal{X}_{norm} - \hat{\mathcal{X}} ||^2$
KL divergence: $L_{KL} = 1/2 \sum(\mu^2 + \sigma^2 - \log(\sigma^2) - 1)$
Total losses:
$L_{total} = L_{GAN} + \lambda^2 \cdot L_{recon} + \lambda^2 \cdot L_{KL}$
Training loop:
For epoch in range (E):
Update Encoder, Generator, Discriminator using L_{total}
Generation:
Sample latent vector: $\mathcal{Z} \sim \mathcal{N}(0,1)$
Generate synthetic data: $\mathcal{X}_{gen} = Generator(\mathcal{Z})$
Return synthetic faulty data $(\mathcal{X}_{gen})$

IX. RESULTS

In this study, an initial subset of 100 randomly selected samples was used to conduct a preliminary evaluation of the proposed synthetic data generation methods. This step aimed to identify the approach capable of producing artificial data most closely resembling the original dataset. The hybrid GAN-VAE model emerged as the most effective, achieving a higher degree of similarity between generated and real data compared (Pearson correlation coefficient of 0.8495) to the individual methods. Following this finding, the entire dataset of 228,622 samples was employed for all subsequent processes, including data balancing, model training, validation, and testing.

The KDE plots of the first six features (Figure 6) confirmed good distributional similarity between the original and generated samples. Next, the VAE method was applied, yielding a lower Pearson correlation coefficient of 0.8232. As shown in Figure 7, KDE plots indicated weaker alignment with the original data distribution compared to GAN.

Finally, a hybrid GAN-VAE model was introduced, combining VAE's ability to capture latent data structure with GAN's adversarial refinement. This approach achieved the best performance, with a Pearson correlation of 0.8796 and excellent KDE distribution similarity (Figure 8). The results demonstrate that GAN-VAE produces more realistic and structurally consistent synthetic data than the standalone models. These results are summarized in Table IX.

TABLE IX. COMPARISON OF DATA BALANCING METHODS

#	Method	Description	Pearson correlation	KDE Distribution similarity
1	GAN	Adversarial training between generator and discriminator; prone to instability	0.8495	Good
2	VAE	Latent coding and reconstruction via probabilistic sampling; less accurate	0.8232	Moderate
3	GAN-VAE	Combines GAN and VAE to generate realistic, balanced, and structurally consistent data	0.8796	Excellent

A. Preprocessing Validation via CNN-based Classification

To validate the effectiveness of the proposed preprocessing and data balancing strategies, a 1D-CNN classification model was trained on the processed dataset. The architecture comprised three convolutional layers, each followed by a ReLU activation function, MaxPooling, and Dropout layers to mitigate overfitting. The convolutional stage concluded with an adaptive median pooling layer, followed by two fully connected layers. The parameters used are presented in Table X. The model achieved a test accuracy of 98.83%, confirming that preprocessing significantly enhanced data quality and improved the discrimination between healthy and faulty sensor states [33-35].

TABLE I. 1D-CNN PARAMETERS

#	Parameter	Value
1	Convolutional layers	3
2	Filters per layer	32 → 64 → 128
3	Kernel size	3
4	Activation function	ReLU
5	MaxPooling size	2
6	Dropout rate	0.4/0.6
7	Batch size	512
8	Number of epochs	50
9	Optimizer	Adam
10	Learning rate	0.00002

Figure 9 presents the results of the classification model trained on the processed and balanced dataset using the methods described in this study. The accuracy curve shows a steady increase, ultimately reaching 98.83%, while the loss curve demonstrates a corresponding decline, indicating stable convergence. The confusion matrix further confirms the model's robustness, with 44,531 fault cases and 43,459 normal cases correctly classified out of a total of 89,066 samples. Misclassifications were minimal, with only 1,074 normal cases incorrectly labeled as faults (False Positives) and just two fault cases mislabeled as normal (False Negatives). These results provide strong evidence of the effectiveness of the preprocessing pipeline.

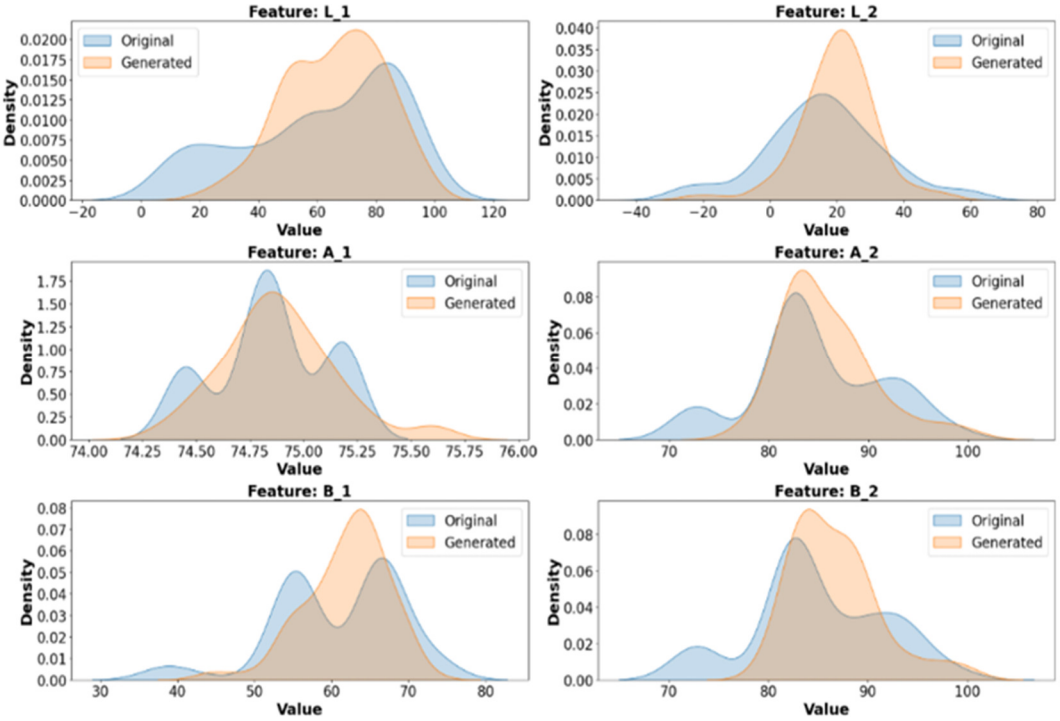

Fig. 6. Pearson correlation and distribution of the first six features using KDE plots for data generation by GAN.

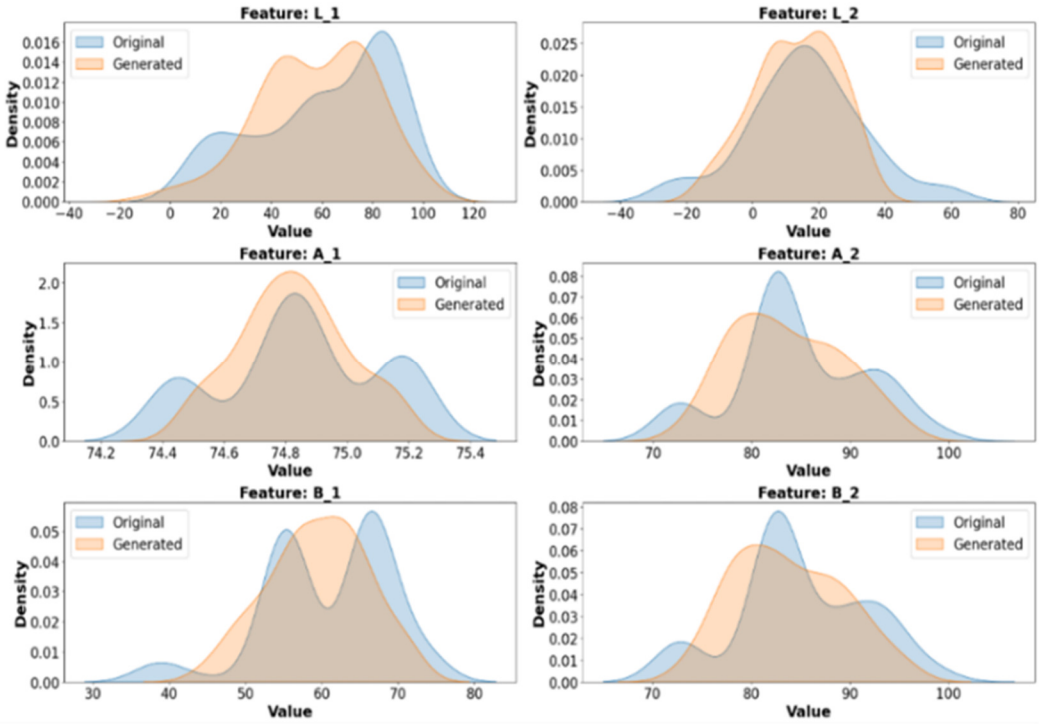

Fig. 7. Pearson correlation and distribution of the first six features using KDE plots for data generation by VAE.

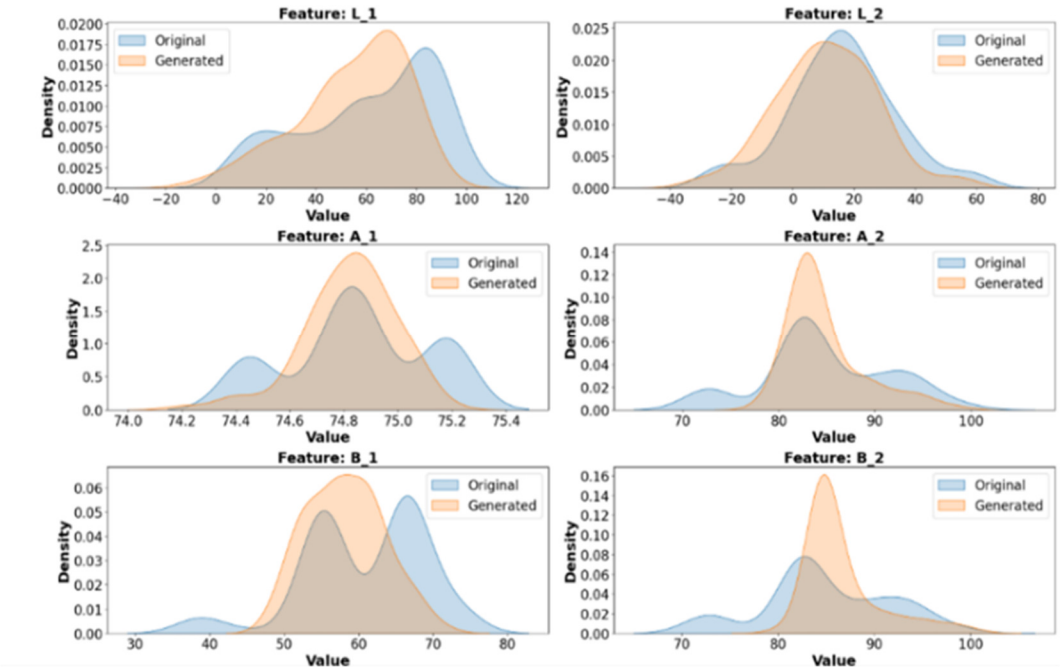

Fig. 8. Pearson correlation and distribution of the first six features using KDE plots for data generation by GAN-VAE.

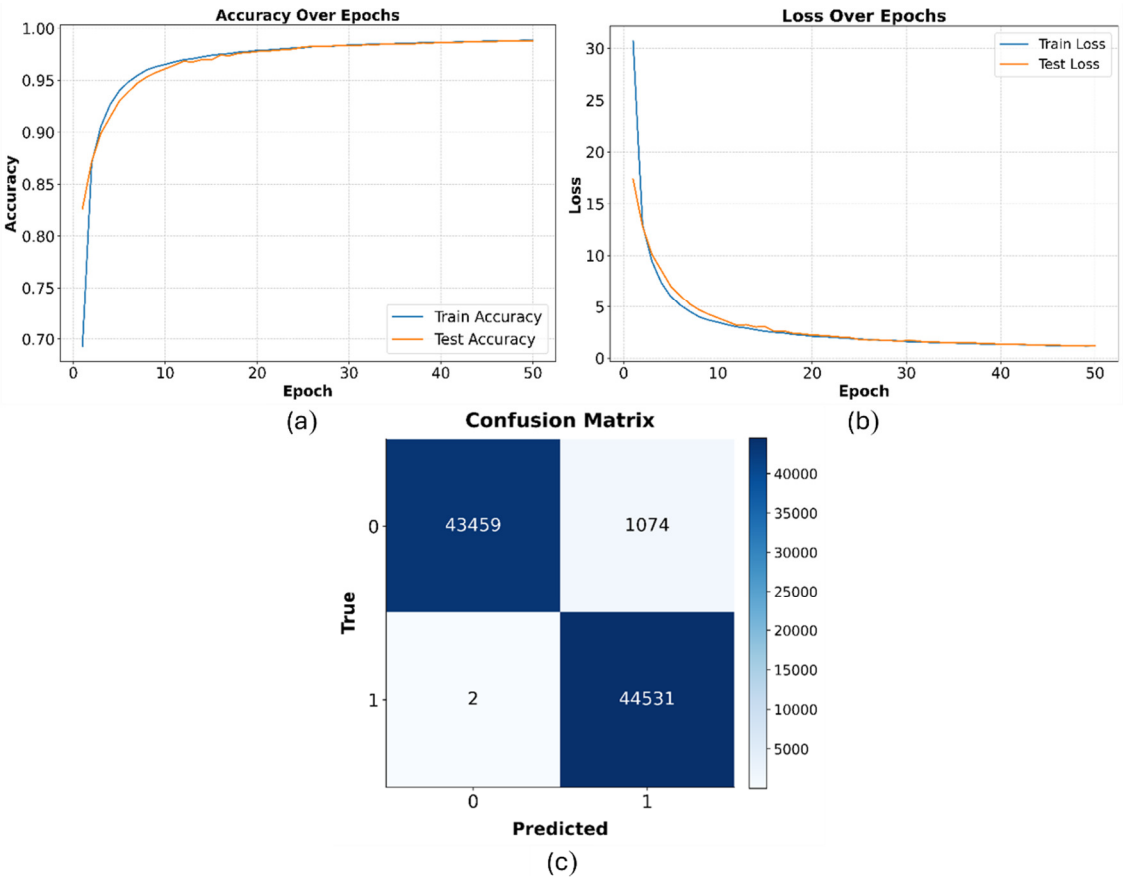

Fig. 9. 1D-CNN model performance: (a) accuracy curves, (b) loss curves, (c) confusion matrix.

X. CONCLUSION

This study proposed an integrated preprocessing and class balancing framework for industrial sensor data. Missing values were addressed, outliers and abnormal values were effectively processed, and class imbalance was mitigated through synthetic data generation. Two generative approaches, Generative Adversarial Network (GAN) and Variational Autoencoders (VAE), were applied individually, and their strengths were further combined into a hybrid GAN-VAE model. Evaluation using Pearson correlation and Kernel Density Estimation (KDE) plots demonstrated that the hybrid approach produced more realistic and structurally consistent synthetic data compared to the standalone models. Finally, the preprocessing pipeline and the GAN-VAE balancing method were validated through a 1-Dimensional Convolutional Neural Network (1D-CNN) classifier, which achieved a test accuracy of 98.83%, confirming the robustness and effectiveness of the proposed methodology in improving data quality and classification performance.

ACKNOWLEDGMENT

This work was completed within the framework of joint academic cooperation and supervision between the authors, each of whom contributed at different stages of research and development.

REFERENCES

[1] F. Duan, S. Zhang, Y. Yan, and Z. Cai, "An Oversampling Method of Unbalanced Data for Mechanical Fault Diagnosis Based on MeanRadius-SMOTE," *Sensors*, vol. 22, no. 14, Jul. 2022, Art. no. 5166, https://doi.org/10.3390/s22145166.

[2] S. B. Belhaouari, A. Islam, K. Kassoul, A. Al-Fuqaha, and A. Bouzerdoum, "Oversampling techniques for imbalanced data in regression," *Expert Systems with Applications*, vol. 252, Oct. 2024, Art. no. 124118, https://doi.org/10.1016/j.eswa.2024.124118.

[3] A. Islam, S. B. Belhaouari, A. U. Rehman, and H. Bensmail, "KNNOR: An oversampling technique for imbalanced datasets," *Applied Soft Computing*, vol. 115, Jan. 2022, Art. no. 108288, https://doi.org/10.1016/j.asoc.2021.108288.

[4] A. Amin *et al.*, "Comparing Oversampling Techniques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study," *IEEE Access*, vol. 4, pp. 7940–7957, 2016, https://doi.org/10.1109/ACCESS.2016.2619719.

[5] Y. Fathy, M. Jaber, and A. Brintrup, "Learning With Imbalanced Data in Smart Manufacturing: A Comparative Analysis," *IEEE Access*, vol. 9, pp. 2734–2757, 2021, https://doi.org/10.1109/ACCESS.2020.3047838.

[6] J. Kafunah, M. I. Ali, and J. G. Breslin, "Handling Imbalanced Datasets for Robust Deep Neural Network-Based Fault Detection in Manufacturing Systems," *Applied Sciences*, vol. 11, no. 21, Oct. 2021, Art. no. 9783, https://doi.org/10.3390/app11219783.

[7] S. Singh and M. T. U. Haider, "Pre-processing of datasets with best feature selection and outlier removal techniques for a fair and robust model of software defect prediction." In Review, May 2022, https://doi.org/10.21203/rs.3.rs-1624790/v1.

[8] O. Celik, M. Hasanbasoglu, M. S. Aktas, O. Kalipsiz, and A. N. Kanli, "Implementation of Data Preprocessing Techniques on Distributed Big Data Platforms," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, Samsun, Turkey, Sep. 2019, pp. 73–78, https://doi.org/10.1109/UBMK.2019.8907230.

[9] M. Zhang, X. Li, and L. Wang, "An Adaptive Outlier Detection and Processing Approach Towards Time Series Sensor Data," *IEEE Access*, vol. 7, pp. 175192–175212, 2019, https://doi.org/10.1109/ACCESS.2019.2957602.

[10] A. Jain, L. Patil, and P. Dandannavar, "Big Data Preprocessing – A Survey of Existing and Latest Outlier Detection Techniques," *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)*, vol. 14, no. 2, pp. 178–181, Apr. 2015.

[11] C. Lartey, J. Liu, R. K. Asamoah, C. Greet, M. Zanin, and W. Skinner, "Effective Outlier Detection for Ensuring Data Quality in Flotation Data Modelling Using Machine Learning (ML) Algorithms," *Minerals*, vol. 14, no. 9, Sep. 2024, Art. no. 925, https://doi.org/10.3390/min14090925.

[12] S. Cofre-Martel, E. L. Droguett, and M. Modarres, "Big Machinery Data Preprocessing Methodology for Data-Driven Models in Prognostics and Health Management," *Sensors*, vol. 21, no. 20, Oct. 2021, Art. no. 6841, https://doi.org/10.3390/s21206841.

[13] A. Hakami, "Strategies for overcoming data scarcity, imbalance, and feature selection challenges in machine learning models for predictive maintenance," *Scientific Reports*, vol. 14, no. 1, Apr. 2024, Art. no. 9645, https://doi.org/10.1038/s41598-024-59958-9.

[14] A. Sharma, P. K. Singh, and R. Chandra, "SMOTified-GAN for Class Imbalanced Pattern Classification Problems," *IEEE Access*, vol. 10, pp. 30655–30665, 2022, https://doi.org/10.1109/ACCESS.2022.3158977.

[15] Q. Liu, G. Ma, and C. Cheng, "Data Fusion Generative Adversarial Network for Multi-Class Imbalanced Fault Diagnosis of Rotating Machinery," *IEEE Access*, vol. 8, pp. 70111–70124, 2020, https://doi.org/10.1109/ACCESS.2020.2986356.

[16] A. Abraham, H. S. Mohideen, and R. Kayalvizhi, "A Tabular Variational Auto Encoder-Based Hybrid Model for Imbalanced Data Classification With Feature Selection," *IEEE Access*, vol. 11, pp. 122760–122771, 2023, https://doi.org/10.1109/ACCESS.2023.3329139.

[17] M. Yin, J. Tian, Y. Wang, and J. Jiang, "A Novel Distributed Process Monitoring Framework of VAE-Enhanced with Deep Neural Network," *Neural Processing Letters*, vol. 56, no. 2, Mar. 2024, Art. no. 118, https://doi.org/10.1007/s11063-024-11577-1.

[18] S. Chatterjee and Y.-C. Byun, "Leveraging generative adversarial networks for data augmentation to improve fault detection in wind turbines with imbalanced data," *Results in Engineering*, vol. 25, Mar. 2025, Art. no. 103991, https://doi.org/10.1016/j.rineng.2025.103991.

[19] Y. S. Hindistan and E. F. Yetkin, "A Hybrid Approach With GAN and DP for Privacy Preservation of IIoT Data," *IEEE Access*, vol. 11, pp. 5837–5849, 2023, https://doi.org/10.1109/ACCESS.2023.3235969.

[20] M. A. Hailan, B. M. Albaker, and M. S. Alwan, "Two-Dimensional Transformation of a Conventional Manufacturer into a Smart Manufacturer: Architectonic Design, Maintenance Strategies and Applications," *Al-Iraqia Journal for Scientific Engineering Research*, vol. 1, no. 1, pp. 77–87, Sep. 2022, https://doi.org/10.33193/IJSER.1.1.2022.39.

[21] A. Fadlil, Herman, and D. Praseptian M, "K Nearest Neighbor Imputation Performance on Missing Value Data Graduate User Satisfaction," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 4, pp. 570–576, Aug. 2022, https://doi.org/10.29207/resti.v6i4.4173.

[22] M. F. S. AlRijeb, M. L. Othman, A. Ishak, M. K. Hassan, and B. M. Albaker, "Machine Learning-Driven Soft Sensor Implementation for Real-Time Fault Detection in CDU of Oil Refinery," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 20425–20432, Feb. 2025, https://doi.org/10.48084/etasr.9781.

[23] P. M. Goad, P. J. Deore, and V. B. Patil, "A novel approach for detecting outliers by using Isolation Forest with reducing under fitting issue." In Review, Dec. 2022, https://doi.org/10.21203/rs.3.rs-2376758/v1.

[24] H. A. H. Al-Najjar, B. Pradhan, R. Sarkar, G. Beydoun, and A. Alamri, "A New Integrated Approach for Landslide Data Balancing and Spatial Prediction Based on Generative Adversarial Networks (GAN)," *Remote Sensing*, vol. 13, no. 19, Oct. 2021, Art. no. 4011, https://doi.org/10.3390/rs13194011.

[25] F. Meghdouri, T. Schmied, T. Gärtner, and T. Zseby, "Controllable Network Data Balancing with GANs," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, Dec. 2021.

[26] A. Von Birgelen, D. Buratti, J. Mager, and O. Niggemann, "Self-Organizing Maps for Anomaly Localization and Predictive Maintenance

in Cyber-Physical Production Systems," *Procedia CIRP*, vol. 72, pp. 480–485, 2018, https://doi.org/10.1016/j.procir.2018.03.150.

[27] D. Zou *et al.*, "Outlier detection and data filling based on KNN and LOF for power transformer operation data classification," *Energy Reports*, vol. 9, pp. 698–711, Sep. 2023, https://doi.org/10.1016/j.egyr.2023.04.094.

[28] E. F. Hadi, M. Z. Bin Baharuddin, and A. W. M. Zuhdi, "Advancing Predictive Maintenance: Median-Based Particle Filtering in MOSFET Prognostics," *Journal Européen des Systèmes Automatisés*, vol. 57, no. 4, pp. 1103–1117, Aug. 2024, https://doi.org/10.18280/jesa.570417.

[29] H. Perez and J. H. M. Tah, "Improving the Accuracy of Convolutional Neural Networks by Identifying and Removing Outlier Images in Datasets Using t-SNE," *Mathematics*, vol. 8, no. 5, Apr. 2020, Art. no. 662, https://doi.org/10.3390/math8050662.

[30] A. Popov *et al.*, "Reduced Graphene Oxide and Polyaniline Nanofibers Nanocomposite for the Development of an Amperometric Glucose Biosensor," *Sensors*, vol. 21, no. 3, Feb. 2021, Art. no. 948, https://doi.org/10.3390/s21030948.

[31] J. E. Choi, D. H. Seol, C. Y. Kim, and S. J. Hong, "Generative Adversarial Network-Based Fault Detection in Semiconductor Equipment with Class-Imbalanced Data," *Sensors*, vol. 23, no. 4, Feb. 2023, Art. no. 1889, https://doi.org/10.3390/s23041889.

[32] S. Stocksieker, D. Pommeret, and A. Charpentier, "Data Augmentation with Variational Autoencoder for Imbalanced Dataset." arXiv, Dec. 2024, https://doi.org/10.48550/arXiv.2412.07039.

[33] J.-H. Lee, J.-H. Lee, C.-J. Lee, S.-L. Lee, J.-P. Kim, and J.-H. Jeong, "A Study on Wheel Member Condition Recognition Using 1D–CNN," *Sensors*, vol. 23, no. 23, Nov. 2023, Art. no. 9501, https://doi.org/10.3390/s23239501.

[34] K. Yan and X. Zhou, "Chiller faults detection and diagnosis with sensor network and adaptive 1D CNN," *Digital Communications and Networks*, vol. 8, no. 4, pp. 531–539, Aug. 2022, https://doi.org/10.1016/j.dcan.2022.03.023.

[35] M. Kassem and B. M. Albaker, "Efficient Classification Model of Pneumonia Infection Based on Deep Transfer Learning and Chest X-Ray Images," *Al-Iraqia Journal for Scientific Engineering Research*, vol. 1, no. 1, pp. 58–67, Sep. 2022, https://doi.org/10.33193/IJSER.1.1.2022.37.